DISKETTE IM HEFT

SONDERHEFT 63 Markt&Technik ÖS 120,-/Sfr. 16,-/Lit. 16000 hfl. 21,-/dkr. 75,-/fmk 62,- DM 16,-

64'er

ÜBER 15 PROGRAMME AUF DISKETTE

GRAFIK

Sprites-Sensation

Text und Grafik mischen ohne Flimmern

Zeichenprogramm

EGA: Zeichenprogramm der Superlative

Intro-Programmierung

3 professionelle Editoren für heiße Intros

Super Tips & Tools

Trickreiche Routinen für Hires und Multicolor

EINER MUSS DEN JOB JA MACHEN

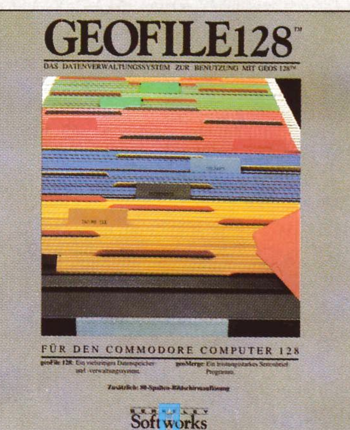
Viele Daten zu haben, ist eine Sache. Diese Daten auch zu finden, wenn man sie gerade braucht, ist leider eine andere. Das muß aber nicht sein. Mit GeoFile können GEOS-Anwender ihren C64 und C128 wieder einmal nutzbringend einsetzen. Damit die Daten nicht nur gesucht, sondern auch gefunden werden. Nicht irgendwann, sondern augenblicklich. Klick, sofort.

In Sekundenschnelle ist ein Formular entworfen. Wenn Sie wissen, wie man mit einer Maus ein Rechteck zieht, können Sie schon eine eigene Datei anlegen. Steht ein Feld zu weit rechts? Oder ist es zu groß geraten? Klicken Sie noch mal, und ändern Sie einfach die Größe oder Position des Feldes. Auch kein Problem mit GeoFile: Formulare bis zum Format DIN A4. Sogar Grafiken können Sie einfügen.

Schwieriger wird's nicht mehr. Steht das Formular, tragen Sie Ihre Daten ein. Müssen Sie sich mit Namen, Adressen und Geburtsdaten herumschlagen? Sich mit Bestellmengen, Preisen und Lieferfristen abgeben? Oder sich mit Gehältern, Steuern und Versicherungsprämien auseinandersetzen? GeoFile hilft Ihnen gerne. Wenn Sie etwas suchen, füllen Sie das Suchformular aus und klicken Sie. Dann sucht GeoFile, was Sie auch immer wissen möchten. Zum Beispiel alle golfspielenden Münchner, die gerne wandern – oder alle Urlaubsorte an der Nordseeküste mit mehr als 10.000 Übernachtungen pro Jahr. Vielleicht möchten Sie ja auch nur wissen, welche Telefonnummer Ihre Bankfiliale hat.

Und GeoFile kann ganz schön Druck machen: auf Computerpapier, Karteikarten oder Adreßaufkleber. Grafische Ausgabe oder Textdruck, und bis zu 16 Layouts pro Datei. Geben Sie die ganze Datei aus, oder nur einen ausgewählten Teilbestand. Mit oder ohne Feldnamen und Umrahmungen. GeoFile druckt, *was* Sie möchten, und zwar so, *wie* Sie es wollen.

Aber das kennen Sie schon von GEOS. Wahrscheinlich überrascht Sie auch dies nicht mehr: Mit GeoFile und GeoMerge werden programmierte Serienbriefe zum Kinderspiel. Sie legen fest, welche Daten einzusetzen sind, und tragen ein, wo und wann dies der Fall sein soll. Eine intelligente Schreibzentrale – GEOS mit GeoFile. Solche tatkräftigen Assistenten kann man sich nur wünschen. Denn wer kümmert sich sonst um Ihre Daten?



GeoFile 64,
Bestell-Nr. 50324
DM 59,–*

GeoFile 128,
Bestell-Nr. 50330
DM 79,–*

* Unverbindliche Preisempfehlung

Markt &Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften und in den Fachabteilungen der Warenhäuser.

Einen 20seitigen farbigen Katalog gibt's unverbindlich bei der Telefon-Hotline 0 21 91/86 61.

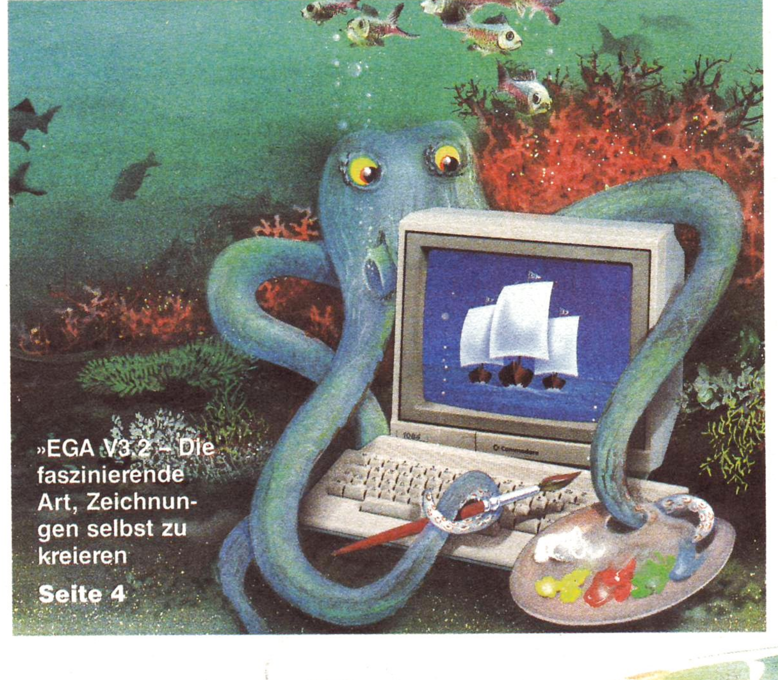


Kluge Köpfe setzen auf GEOS



2315/008





»EGA V3.2 – Die faszinierende Art, Zeichnungen selbst zu kreieren
Seite 4

Zeichenprogramm

EGA V3.2 – für Grafikkünstler
Ein Zeichenprogramm der Superlative. Auf vier Bildschirmseiten wird selbst der letzte Speicherwinkel ausgenützt. **4**

Intros

IRQ-Designer – tolle Unterbrechungen
Interrupt-Programmierung ohne große Programmierkenntnisse – für Ihre Vorspänne **9**

Stardemomaker – neue Dimensionen
Fantastische Intros und Vorspänne lassen sich durch einfache Menüführung gestalten **14**

Movememt-Editor – Bewegung wie im Film
Kreieren Sie faszinierende Sprite-Balletts für Ihre Spiele oder Intros **15**

Diashow

Die besten Colorbilder
Eine Auswahl der besten Grafiken aus den Wettbewerben der letzten Jahre **17**

Tips & Tricks

Deluxe Datas – der Datenexpreß
Maschinenprogramme und Grafikbilder im Handumdrehen in Basic-Zeilen umgewandelt **22**

Color is easy
Unentbehrliche Tools für alle denkbaren Bereiche der Grafik und die komplette Registerbelegung des VIC **23**

B.M.G – Maskengenerator für Verwöhnte
Der Entwurf von Bildschirmmasken wird zum Kinderspiel: Der komfortable Editor wandelt sie zusätzlich in Basic-Zeilen. **28**

Text und Grafik ohne Flimmern mischt »Tegra«, eine komfortable Befehlserweiterung
Seite 40

Die schönsten Grafikbilder am laufenden Band finden Sie in unserer Dia-Show Seite 17

Anwendung

Funktionenhilfe – rasante Kurven
Berechnen und plotten Sie einfach und professionell Funktionen mit Extremwerten und Definitionslücken **29**

Zeichensatz

Grundlagen der Programmierung
Wir starten in die Wunderwelt der Grafik des C64 mit einem ausführlichen Kurs über Fonts **34**

Sprites

Tegra – nichts ist unmöglich
Text und Grafik gemischt auf einem Bildschirm. Die außergewöhnliche Grafikerweiterung stellt Ihnen zusätzlich ein Zeichenprogramm zur Verfügung. **40**

View Basic – Geisterstunde
Mit diesem Sprite-Editor der Extraklasse kommt kein Frust auf beim Konstruieren von Sprites **44**

Programmiergrundlagen für Sprites
Erfahren Sie in diesem Kurs, wie einfach sich Sprites konstruieren lassen **47**

Sonstiges

Impressum **20**

Disklader **21**

Leserumfrage **38**

Wettbewerb **46**

Vorschau **50**

Alle Programme aus Artikeln mit einem [Disketten]-Symbol finden Sie auf der beiliegenden Diskette (Seite 19)

EGA V3.2 – neue Grafikdimensionen

Für Grafik-künstler

Auch der hinterste Speicherwinkel des C64 wird vom Zeichenprogramm »EGA V3.2« optimal genutzt: Vier Grafikseiten lassen sich gleichzeitig damit bearbeiten.

von Sönke Meyer und Thomas Lipp

Dieses Zeichenprogramm der Superlative besitzt eine Fülle von Funktionen, die jede gewünschte Grafik auf den Bildschirm zaubern (Abb. 1).

Auf der Diskette zum Sonderheft finden Sie allerdings zwei Versionen: für den »echten« C64 und den C-64-Modus des C128.

Laden Sie das für Ihre Gerätekonfiguration passende Programm mit:

```
LOAD "EGA V3.2 C 64",8
```

bzw. LOAD »EGA V3.2 C128«,8 und starten es mit RUN. **Egal, welche Version Sie benutzen: Verwenden Sie nur die genannte Ladeanweisung (Direkteingabe oder aus dem Disketteninhaltsverzeichnis), nicht den Disklader! Erst wenn Sie die für Ihren Computer passende Programmversion in »EGAKOMP« umbenannt haben (RENAME-Funktion der Floppy, s. Handbuch), kann man sie auch mit der Benutzeroberfläche in den Computer holen.**

Durch Tastendruck gelangt man vom Titelbild in den Arbeitsbildschirm. Links oben erscheint die Koordinatenanzeige für die x-Richtung, links unten der vertikale Wert. In der Mitte des oberen Bildschirmrandes sehen Sie vier Rechtecke: Der jeweils aktive Grafikbildschirm ist hier revers gekennzeichnet. »Electronic Graphite Abundance«, abgekürzt EGA, arbeitet mit einer Zeichenfläche von insgesamt 640 x 400 Bildpunkten, auf vier Bildschirme von je 320 x 200 Pixel verteilt.

Der untere Bildschirmrand zeigt einen gelben Balken unter dem Hinweis »SPEED«. Gemeint ist damit die variable Geschwindigkeit des Grafik-Cursors. Sie ist zu Programmbeginn auf Maximum eingestellt: Der Balken besitzt die volle Länge. Darunter steht der Hinweis auf die aktuelle Zeichenfunktion, voreingestellt ist »DRW« (für Draw = Zeichnen). Ganz rechts auf dem unteren Bildschirmrand kann man neben dem Wort »FILL« ein Sprite erkennen, das das momentan gültige Füllmuster enthält. Bleibt noch das Sprite im Bildschirmrand rechts oben: Es besteht aus dem Buchstaben »M« und zwei Strichen, die den Winkel anzeigen, in dem mit der MOV-Funktion »fotografierte« Grafikausschnitte abgelegt werden. Der Ausschnitt wird exakt in der Richtung eingeklemmt, welche die beiden Striche definieren, ausgehend von der aktuellen Position des Grafik-Cursors. Als Zeichengerät fungiert der Joystick in Port 2.

Die Funktionen des Programms gliedern sich in Haupt- und Nebenfunktionen, als Kürzel in der Mitte unten zeigt »EGA« nur die Hauptbefehle an. Bis auf »ZOOM« lassen sich alle Funktionen mit der STOP-Taste abbrechen, man kommt dann automatisch in den DRAW-Modus. Da bei manchen Funktionen (z.B. LINE) ein zweites Grafik-Cursor-Kreuz aktiviert wird, ist beim Bildschirmwechsel dieser Cursor nicht mehr sichtbar und erscheint erst wieder bei der Rückkehr zum Ausgangsbildschirm.

Haben Sie eine Fehleingabe gemacht, oder ist eine gewählte Funktion nicht ausführbar, brummt der Computer vernehmlich. Richtige Eingaben werden bei manchen Befehlen mit einem hohen Klangsignal belohnt. Nahezu alle Zeichenmodi lassen sich über alle vier Bildschirme hinweg ausführen, die entsprechende Lage der Bitmaps im C64 zeigt unsere Tabelle.

Die Funktionen beider EGA-Versionen sind identisch. Sie erreichen sie mit folgenden Tasten (in Klammern der damit aktivierte Modus):

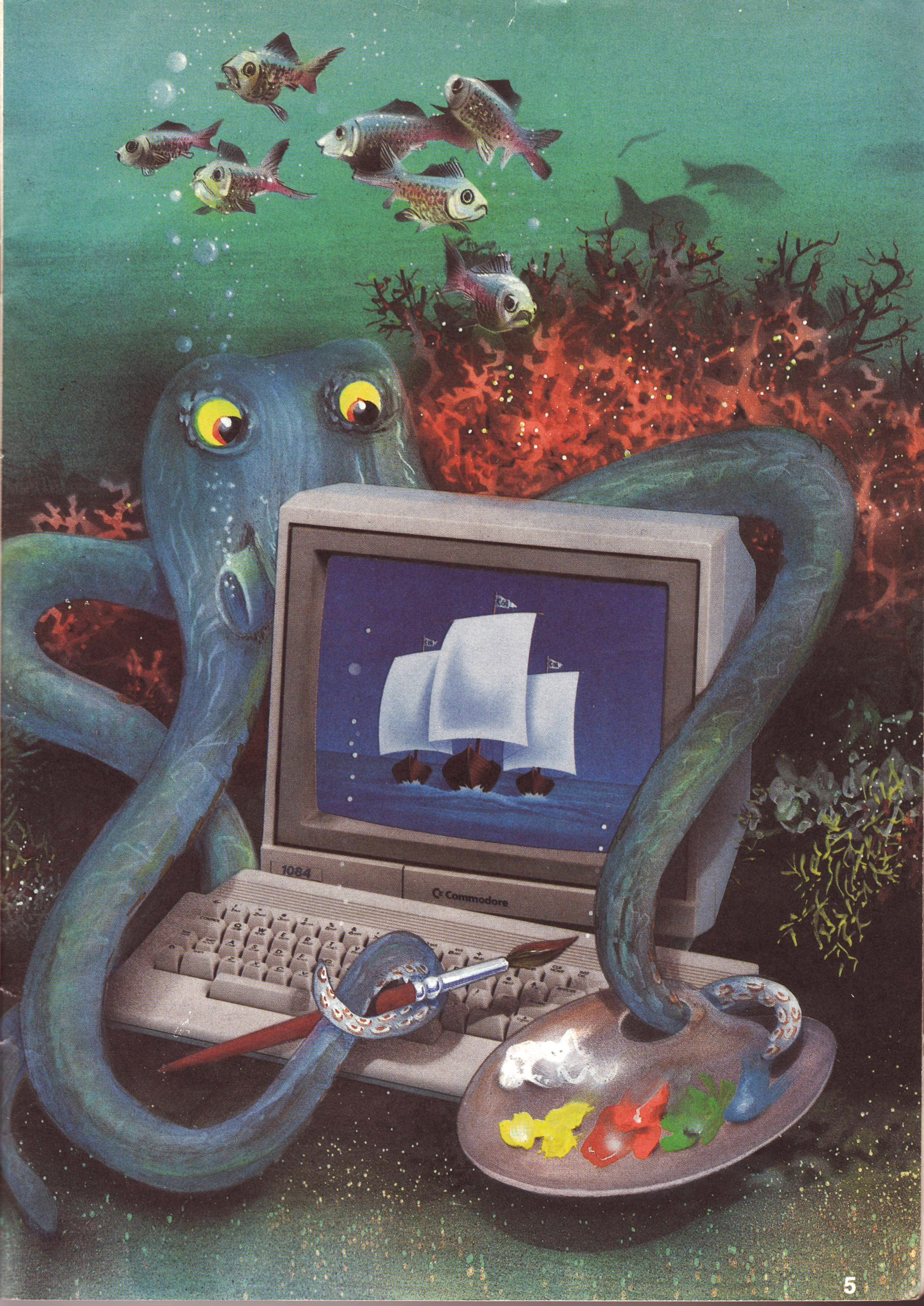
<SPACE> (ZOM): »Zoomen« bedeutet »vergrößern«, d.h., ein 40 x 25 Pixel großer Ausschnitt wird vergrößert dargestellt. Das Sprite zur Anzeige des aktuellen Bildschirms verschwindet, erst nach Verlassen des Zoom-Modus durch erneuten Druck auf die Leertaste wird es wieder aktiviert. Per Joystick können Sie das Cursor-Kreuz durch den gesamten, 640 x 400 Punkte großen Bereich scrollen.

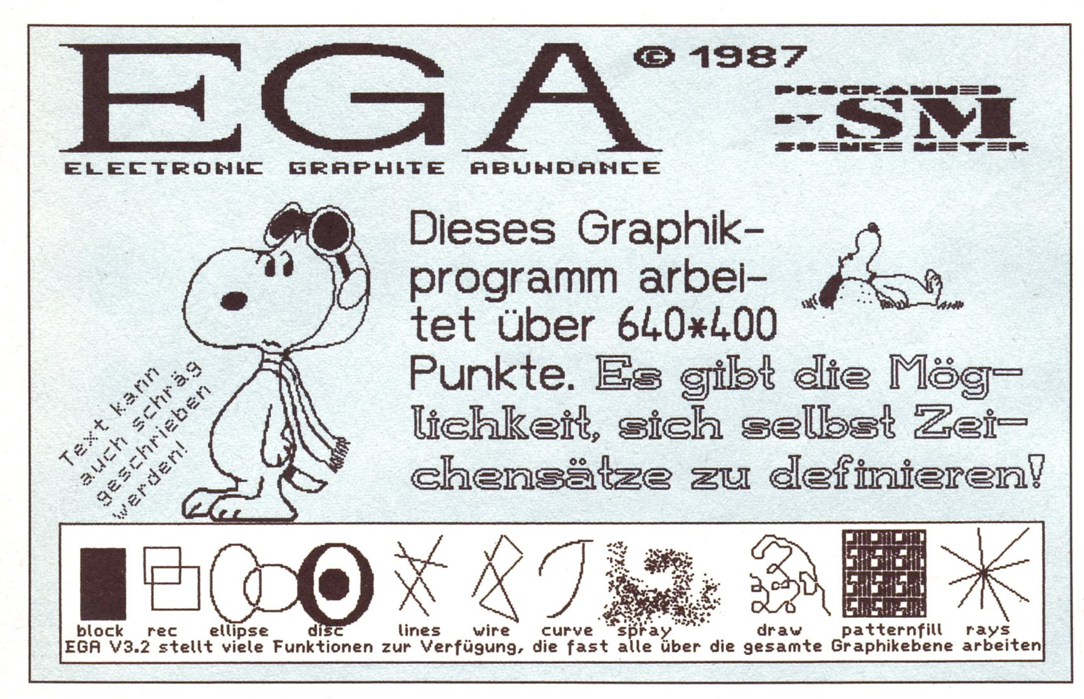
<L> (LIN): Linie zeichnen. Dazu bewegt man den Grafik-Cursor zum Ausgangspunkt der gewünschten Linie und drückt den Feuerknopf. Bewegen Sie nun das Kreuz mit dem Joystick zum gewünschten Endpunkt der Linie und betätigen Sie erneut den Feuerknopf.

<W> (WIR): Wire = Draht. Diese Funktion gestattet das Zeichnen mehrerer Linienzüge. Nachdem der Startpunkt per Feuerknopf festgelegt wurde, zeichnet dieser Modus jeweils eine Linie zum neuen Endpunkt, ausgehend vom letzten.

<R> (RAY): Ray = Strahl. Nach der Auswahl eines Fixpunktes zieht das Programm Linien bis zur aktuellen Position des Grafik-Cursors. So erhalten Sie ein Strahlennetz. Wenn Sie den Feuerknopf gedrückt halten und den Joystick-Hebel kreisförmig bewegen, entsteht eine typische Moiré-Grafik (Abb. 2).

<E> (ELL): Ellipse. Will man eine Ellipse oder einen Kreis zeichnen, muß mit dem Grafikkreuz der Mittelpunkt festgelegt werden. Ziehen Sie anschließend ein gepunktetes Rechteck auseinander, dessen Entfernung der Seiten vom Mittelpunkt mit den Radien identisch ist. Dafür steht ein Maximalwert von 255 Bildpunkten zur Verfügung. Für einen Kreis müssen die Seitenlängen in x- und y-Richtung gleich sein. Teile einer Ellipse oder eines Kreises, die über den Bereich


1084
Commodore



◀ **[1] Vielseitigkeit ist die Stärke von EGA**

Nach jedem Druck auf den Feuerknopf wird rechts unterhalb des Cursors ein 16 x 16 Pixel großer Bereich aus dem Grafikbild in die Füllmusteranzeige übertragen. Das Füllmuster läßt sich scrollen, wenn Sie den Joystick bei gedrücktem Feuerknopf bewegen.

<*> **(SPR):** Sprayfunktion. Bei gedrücktem Feuerknopf werden rund um den Grafik-Cursor Punkte verstreut.

<**A**> **(AND):** UND-Verknüpfung. Zwei Grafikseiten mit jeweils 320 x 200 Punkten lassen sich UND-verknüpfen (nur wenn in beiden Grafiken zwei Bildpunkte an der gleichen Position eingeschaltet sind, werden diese im UND-verknüpften Bild berücksichtigt). Nach Befehlsaktivierung beginnt der Ausschnitt zu blinken, in dem sich der Cursor befindet. Mit <CRSR rechts> kann man das blinkende Viertel weiterschalten. Nach Druck auf die RETURN-Taste erfolgt die logische UND-Verknüpfung beider Grafikseiten.

des Bildschirms hinausgehen, werden abgeschnitten (gilt auch für andere Funktionen).

<**C**> **(CUR):** Um Ecken abzurunden, eignen sich sehr gut Kurven oder Viertelellipsen. Dazu muß man so vorgehen: den Grafik-Cursor an einem beliebigen Mittelpunkt positionieren, den Feuerknopf drücken und das dabei entstehende gepunktete Rechteck in folgenden Richtungen auseinanderziehen:

1. nach links oben (Kurve links oben abgerundet),
2. nach rechts oben (Viertelellipse rechts abgerundet),
3. nach links unten (Kurve links unten ausgebuchtet),
4. nach rechts unten (Teilellipse nach rechts unten abgerundet).

<**D**> **(DSC):** Disc, ausgefüllte Ellipse. Die Funktionsweise ist identisch mit ELL <E>, allerdings werden Kreise und Ellipsen ausgefüllt (Abb. 3).

<**Q**> **(QUAD):** Quadragle, Rechteck. Um rechteckige Gebilde zu zeichnen, muß man wie bei LIN zwei gegenüberliegende Punkte fixieren (am besten in vertikaler Richtung versetzt), die den oberen und unteren Eckpunkt des Rechtecks bilden.

<**B**> **(BLK):** Block, wie QUAD, das Rechteck wird allerdings ausgefüllt.

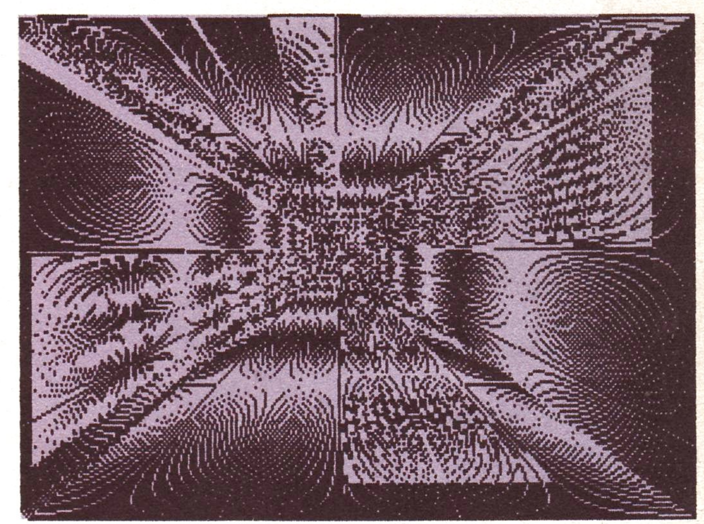
<**F**> **(FIL):** Fill, Füllen. Diese Funktion füllt eine begrenzte Fläche aus. Sie darf sich über vier Teilbildschirme erstrecken. Diese Funktion wird durch erneuten Druck auf den Feuerknopf deaktiviert.

<**P**> **(PFL):** Pattern Fill, Füllen mit Mustern. Damit wird eine umgrenzte Fläche mit dem im unteren Bildschirmbereich angezeigten Muster ausgefüllt. Dieser Befehl wirkt sich jeweils nur auf den aktiven Grafikbildschirm aus. Falls der Zeichensatz im Speicher zu umfangreich ist, bricht diese Funktion ab. Dann muß der Zeichensatz auf Diskette ausgelagert und dessen Speicher im Computer mit <SHIFT K> gelöscht werden. Mit der Kommataste <,> holen Sie die Grafik zurück.

<**SHIFT P**> **(DTP):** Define Pattern, Füllmuster gestalten.

Kurzinfo: EGA V3.2

Programmart: Zeichenprogramm im Hires-Modus
Laden: LOAD "EGA V3.2 C64",8 (nur für C64)
LOAD "EGA V3.2 C128",8 (nur für C128 im C-64-Modus)
Starten: nach dem Laden der entsprechenden Version mit RUN
Steuerung: Tastatur/Joystick Port 2
Besonderheiten: Vier Grafikseiten mit einer Gesamtauflösung von 640 x 400 lassen sich bearbeiten.
Benötigte Blocks: 111 (Hauptprogramm, Zeichensätze, Druckertreiber)
Programmautoren: Sönke Meyer, Thomas Lipp



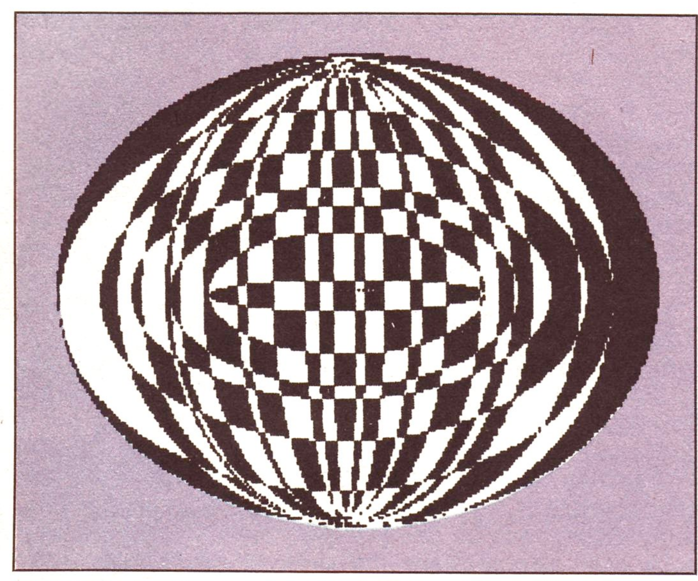
[2] Moiré-Grafik mit der Funktion RAY im invertierenden Modus, eingestellt mit Taste <F7>

<**O**> **(ODR):** ODER-Verknüpfung. Dieser Modus ist identisch mit AND zu bedienen, allerdings zeigt der Gesamtbildschirm jetzt alle eingeschalteten Bildpunkte, egal, ob sie in der ersten **oder** zweiten Grafikseite aktiviert waren.

<**X**> **(XOR):** EXKLUSIV-ODER-Verknüpfung. Der Unterschied zu den beiden letzten vorher genannten Modi besteht darin, daß jetzt im neuen Gesamtgrafikbild nur die Bildpunkte gesetzt werden, die sich in beiden Grafikseiten unterscheiden (in der einen gelöscht, bei der anderen aktiv). Übereinstimmende Bit-Inhalte (**beide** gesetzt oder gelöscht) werden immer negativ interpretiert, d.h., das Pixel in der Gesamtgrafik an entsprechender Position wird **nicht** aktiviert.

<←> **(EXQ):** Exchange Screens, Austausch zweier Grafikseiten. Der Ablauf geschieht wie bei den Verknüpfungsmodi.

<**M**> **(MOV):** Move, bewegen. Dies ist die wichtigste Funktion von »EGA«. Sie bildet auch die Grundlage für den Textmodus. Durch den ersten Druck auf den Feuerknopf legt man



[3] Solche Grafiken zaubert die DSC-Funktion

eine Ecke des Bildausschnitts fest, der bewegt werden soll. Ziehen Sie nun ein Rechteck auseinander (maximal 255 x 255 Pixel). Das MOV-Objekt muß in diesem Bereich enthalten sein. Danach drücken Sie den Feuerknopf, jeder weitere Druck darauf legt den Bildschirmausschnitt ab. Ertönt der Fehlergong, war im Zeichensatzspeicher nicht genügend Platz. Mit der Tastenkombination <SHIFT T> läßt sich der Drehwinkel in 45-Grad-Schritten ändern, mit dem MOV-Objekte abgelegt werden.

<Z> (TXT): Textmodus. Bevor Sie ihn einschalten, muß ein Zeichensatz geladen werden. Drei Beispiele befinden sich auf der Diskette zum Sonderheft: Zeichensatz 1 bis 3 (Abb. 4), mit der Tastenkombination <SHIFT L> läßt sich der Ladevorgang einleiten. Unmittelbar nach Aktivierung dieses Modus sitzt der Grafik-Cursor fest. Der Computer reagiert nur noch auf die Tasten <CTRL 9>, <CTRL 0>, <CTRL K>, <RETURN>, <DEL> und <RUN/STOP>. Damit nach einem etwaigen Löschen die Grafik restauriert wird, empfiehlt sich, im »invertierenden« Modus (F7) zu schreiben. Aus programmtechnischen Gründen läßt sich immer nur die Zeile editieren, in der man sich befindet. Mit <RETURN> beginnt eine neue Zeile. Der Zeilenabstand richtet sich nach dem höchsten in der letzten Zeile enthaltenen Zeichen.

<SHIFT Z> (DCH): Define Character, Zeichen definieren. Dieser Befehl funktioniert zu Beginn wie MOV, allerdings wird nach Auswahl des Grafikausschnitts diesem ein ASCII-Code zugeordnet. Drücken Sie also nach der Definition des Ausschnitts die Taste oder Kombination, nach der dann im Textmodus der Grafikausschnitt erscheinen soll. Ein hoher Gong ertönt. Alle Tastaturzeichen und Kombinationen sind erlaubt, die auch der Editor des Basic 2.0 annehmen würde. Ausnahmen: <CTRL 9>, <CTRL 0>, <RETURN> und <DEL>. Ist der tiefe Fehlerton zu hören, liegt es daran, daß der Zeichensatzspeicher voll ist. Gehen Sie in diesem Fall vor, wie bei »Pattern Fill (PFL)« beschrieben.

<SHIFT K> (KIL): Kill Character Memory. Löscht den Zeichensatzspeicher.

<SHIFT E> (ECH): Erase Character. Ein einzelnes Zeichen wird gelöscht.

<SHIFT C> (CHA): Change, Ändern des ASCII-Code. Einem gewählten Zeichen wird eine neue Tastenkombination zugewiesen.

<SHIFT U> (UCH): Use Character, Zeichen verwenden. Nach Auswahl eines vorhandenen Zeichens läßt sich wie bei MOV der Cursor frei bewegen. Druck auf den Feuerknopf legt das Zeichen ab. Damit kann man z.B. eine Pinselfunktion simulieren.

<SHIFT M> (MEM): Display Memory, Speicher anzeigen. Der Umfang des noch frei verfügbaren Speicherplatzes erscheint auf dem Bildschirm. Dies ist auch für MOV wichtig, da alle MOV-Objekte im freien Bereich zwischengespeichert werden.

<SHIFT S> (ZSV): Zeichensatz sichern. Der aktuelle Zeichensatz im Speicher wird auf Diskette gespeichert. Ein entsprechender File-Name ist anzugeben.

<SHIFT L> (ZLO): Zeichensatz laden. Unmittelbar nach Aktivierung dieser Funktion liest der Computer das Disketteninhaltsverzeichnis. Mit <CRSR aufwärts/abwärts> läßt sich der gewünschte File-Name des Zeichensatzes auswählen, <RETURN> löst den Ladevorgang aus.

<CBM S> (GSV): Grafik sichern. Nach Abfrage, ob eine Einzelgrafik (aktueller Bildschirm) oder die gesamten vier Seiten gespeichert werden sollen, muß man einen File-Namen eingeben. Das Programm sichert die Grafiken im Hi-Eddi-Format (320 x 200 Punkte). Bei der Wahl der »Gesamtgrafik« speichert der Computer trotzdem vier einzelne Bilder.

<CBM L> (GLO): Grafik laden. Dies geschieht wie bei einem Zeichensatz. Bei »Gesamtgrafik« bleibt noch immer die



[4] Unterschiedliche Zeichensätze auf einer Grafikseite

Möglichkeit, das Bild als Einzelgrafik in den Computer zu holen. Diese wird aber grundsätzlich in den aktuellen Bildbereich geladen.

<CBM C> (CMD): Command, Floppybefehle senden.

Damit können Sie einen der üblichen DOS-Befehle ans Laufwerk schicken, ohne OPEN und CLOSE (es genügt wie gewohnt auch die Befehlsabkürzung z.B. »N« statt »NEW« zum Formatieren, »I« für »INITIALIZE«, »S« zum Löschen usw.).

< . > (HSP): Hilfsspeicher. Grafiken lassen sich dort hineinkopieren. Ist der Zeichensatz zu groß, wird diese Funktion mit einem akustischen Hinweis abgebrochen.

Die Hilfsfunktionen

Gewisse Tastenfunktionen sind nur in bestimmten Modi zulässig. Manche arbeiten nur dann problemlos, wenn der Vorrat an restlichem Speicherplatz dies noch zuläßt:

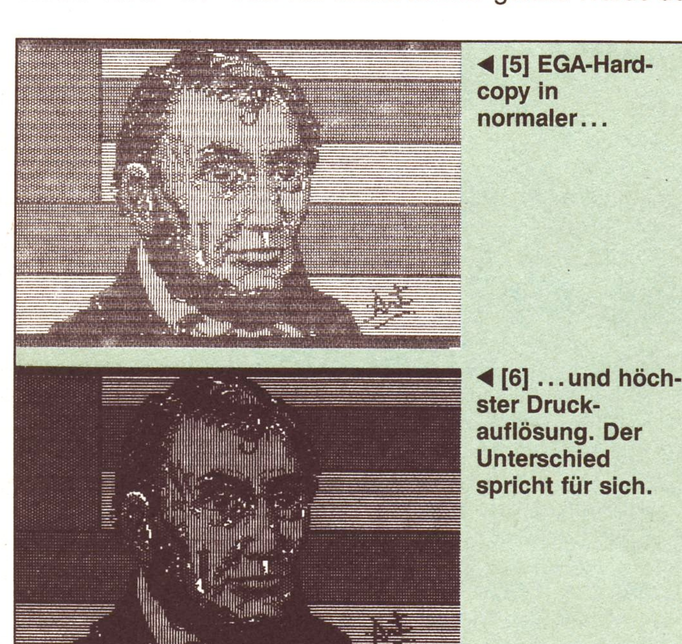
<CTRL 9>: Revers-Modus für MOV-Objekt und Textmodus einschalten.

<CTRL 0>: Revers-Modus deaktivieren.

<CTRL K>: ermöglicht die Ausgabe von Text und MOV-Objekten in kursiver Form. Beim ersten Aufruf dieser Hilfsfunktion entsteht eine Neigung der Figur, ein weiterer Druck halbiert sie. Drücken Sie die Tastenkombination ein drittes Mal, wird wieder der normale Modus eingeschaltet.

<SHIFT T>: Diese Funktion dreht den Winkel, in dem Text und MOV-Objekte abgelegt sind, um 45 Grad im Uhrzeigersinn.

<SHIFT CLR/HOME>: Löschen der aktuellen Grafikseite, im Zoom-Modus betrifft dies nur das sichtbare 40 x 25-Pixelfeld.
<I>: invertiert den sichtbaren Bildschirm.
<S>: Der Inhalt der Grafikseite wird in x- oder y-Richtung gestreckt.
<T>: spiegelt die aktuelle Grafikseite um die x- und y-Achse.
<+>: erhöht die Geschwindigkeit des Grafik-Cursors.
<–>: verringert dessen »Speed«.
<F1>: ändert die Farbe des Grafik-Cursor-Sprite.
<F3>: Die Zeichenfarbe (Vordergrund) wird erhöht.
<F5>: paßt die Hintergrundfarbe Ihren Wünschen an.
<F7>: Damit läßt sich der Zeichenmodus auswählen. Zur Verfügung stehen: Punkt invertieren, löschen oder setzen. Alle Zeichenfunktionen arbeiten dann im gewählten Modus. Sie erkennen dies an den Farben des Bildschirmrahmens: Hellblau = Invertieren, Grün = Setzen, Violett = Löschen.
<SHIFT 1 bis 9>: Die Cursor-Position wird der angegebenen Zahl zugeordnet.
<1 bis 9>: Cursor setzt sich an voreingestellte Position.
<,>: Falls der Hilfsspeicher bereits belegt ist, wird er mit der aktuellen Grafikseite getauscht. Der Hilfsspeicher wird mit der Taste <.> oder mit einem Muster gefüllt. Wurde der



◀ [5] EGA-Hardcopy in normaler...

◀ [6] ...und höchster Druckauflösung. Der Unterschied spricht für sich.

Inhalt in der Zwischenzeit durch Laden oder Definieren eines Zeichensatzes oder Aufruf der MOV-Funktion überschrieben, bricht die Funktion mit der akustischen Fehlermeldung ab.
<CBM P> Drucken: Nach Wahl dieser Funktion fragt Sie das Programm, ob eine Einzel- oder Gesamtgrafik gedruckt werden soll. Anschließend lädt das Programm die Druckerroutine »EGA-PRINT« und druckt die Grafik auf Epson-Druckern oder -kompatiblen aus.

Auf der Programmdiskette gibt es zwei Dateien (EGA-PRINT.GEN und EGA-PRINT.GEN+), die wie ein Basic-Programm geladen und mit RUN gestartet werden. Beide Routinen schreiben das kompakte Maschinensprache-File »EGA-PRINT« auf Diskette, das exakt unter diesem File-Namen nach Betätigung der Tasten <CBM P> vom Hauptprogramm »EGA« nachgeladen wird und während des Programmablaufs im Bildschirmspeicher ab $0400 (1024) bleibt.

Achtung: Die Assembler-Datei »EGA-PRINT«, die von »EGA-PRINT.GEN« erzeugt wird, befindet sich bereits auf der Diskette zum Sonderheft (das Generatorprogramm muß also nicht mehr geladen und gestartet werden!). Noch nicht installiert, dafür aber um so interessanter ist die Routine »EGA-PRINT.GEN+«. Sie schafft ebenfalls einen Treiber für Epson- und kompatible Drucker, allerdings mit einer Punktdichte von 1920 Punkten pro Zeile. Die dabei erreichte Auflösung auf dem Papier beträgt nun 1920 x 400 (statt 640 x 400) Pixel. So entstehen total schwarze Flächen und vollständige Linien. Der Unterschied ist erheblich: Abb. 5 zeigt die niedrige Auflösung und Druckqualität mit der herkömmlichen Hardcopy-Routine, Abb. 6 dagegen die höhere Auflösung und bessere Qualität mit dem neuen Druckertreiber.

1. Sie laden »EGA-PRINT.GEN+« wie jedes normale Basic-Programm (,8), starten es aber noch nicht mit RUN.

2. Da das Generierungsprogramm ebenfalls eine Datei »EGA-PRINT« erzeugt, müssen Sie das sich bereits darauf befindliche File umbenennen:

```
OPEN 1,8,15,"R:EGA - PRINT.OLD=EGA - PRINT": CLOSE1
```

3. Starten Sie jetzt das bereits geladene Installationsprogramm mit RUN, der nun gültige Druckertreiber für die höhere Auflösung wird auf Diskette geschrieben. Es sind darauf noch exakt die drei Blocks frei, die man dazu benötigt.

Falls Sie doch eines Tages wieder Lust verspüren, die »alte« Routine mit der geringeren Auflösung zu benutzen, kann dies wieder mit dem RENAME-Befehl des Floppy-DOS geschehen. Folgende Direkteingaben sind dazu nötig:

```
OPEN 1,8,15,"R:EGA - PRINT.HIGH=EGA - PRINT"
PRINT#1,"R:EGA - PRINT=EGA - PRINT.OLD": CLOSE1
```

Um später wieder mit der besseren Routine zu arbeiten, muß man den umgekehrten Weg gehen:

```
OPEN 1,8,15,"R:EGA - PRINT.OLD=EGA - PRINT"
PRINT#1,"R:EGA - PRINT=EGA - PRINT.HIGH": CLOSE1
```

Wenn Ihnen diese Befehlsfolgen zu umständlich sind (man kann sie auch als Basic-Programm mit Zeilennummern abtippen und als einzelne Batch-Dateien speichern, z.B. die erste als »BATCH.EGA-LOW«, die zweite als »BATCH.EGA-HIGH«), oder Sie Angst vor Datenverlust haben, ist auf jeden Fall zu empfehlen, zwei separate Disketten für das Ihrer Computerkonfiguration (C 64 oder C-64-Modus) entsprechende Programmpaket anlegen, das unbedingt aus folgenden Dateien bestehen muß:
- EGA V3.2 (passende Version für Ihren Computer),
- EGA-PRINT (für die gewünschte Druckerauflösung),
- ZEICHENSATZ 1
- ZEICHENSATZ 2
- ZEICHENSATZ 3

Schon allein, weil die Diskette zum Sonderheft »knallvoll« ist und keine anderen Daten mehr, vor allem keine selbstgemalten Grafikseiten, darauf gespeichert werden können, sollten Sie so vorgehen.

Ob Sie normale Grafikbilder, umfangreiche Poster oder überdimensionale Bilder mit »EGA« schaffen möchten, überlassen wir Ihrer Fantasie. Der Komfort und die Fülle der Befehle wird jeden Grafikfan begeistern. (gr/bl)

Speicheraufteilung EGA V3.2

Bereich	Inhalt
$E000 bis $FFFF	sichtbare Bitmap
$D000 bis $DFFF	Directory-Speicher
$CC00 bis $CFFF	Farb-RAM
$AC00 bis $CBFF	Bitmap
$8C00 bis $ABFF	Bitmap
$6C00 bis $8BFF	Bitmap
$4600 bis $6BFF	Programmspeicher
$2600 bis $45FF	Hilfsspeicher
$0800 bis $45FF	Zeichensatz- und MOV-Objektspeicher
$0400 bis $07FF	Bildschirm

Nur durch geschickte Speicherplatzaufteilung wird EGA so leistungsfähig

Farbenpracht mit dem IRQ-Designer

Tolle Unterbrechungen

Nur ausgefuchste Profis beherrschten bislang die komplizierte Programmierung des Raster-Interrupts – Bis jetzt! Mit dem IRQ-Designer schaffen Sie im Nu die farbigsten Vorspänne für Ihre Programme.

von Marcus Hillebrand

Der IRQ-Designer entstand, um Raster-Interrupts einfach und schnell programmieren zu können. In erster Linie, um komfortabel sog. Intros zu programmieren. Aber er kann viel mehr:

Bei Lores ist für jede zweite Bildschirmzeile eine andere Farbe programmierbar. Damit lassen sich Effekte erzielen, wie man sie bisher nur von kommerziellen Programmen kannte.

Das Programmpaket besteht aus zwei Hauptteilen:

1. Der High-Speed-Interpreter

Sein Zweck ist die Abarbeitung der mit dem »Editor« (s.u.) aufgestellten Tabellen. Er braucht dazu extrem wenig Zeit (ca. 1½ Rasterzeilen für Decodierung und das Ausrechnen der Befehlsposition)! Erst durch den Interpreter wird es möglich, die Tabellen des Editors als Rasterzeilen-IRQs abzuarbeiten.

Achtung:
Bearbeiten Sie zuerst einen Datensatz mit dem »Editor«, bevor Sie einen der Interpreter laden.

Auf der beiliegenden Diskette befinden sich drei Versionen des High-Speed-Interpreters:

IRQ $CD00 – Basisadresse = $CD00 /52380.
Speicherbedarf: $CD00 - $CFFF

Diese Version ist für eigenständige Intros oder zur Einbindung in Maschinenprogramme gedacht. Geladen wird von der beiliegenden Diskette mit:

```
LOAD"IRQ - $CD00",8,1
```

Beachten Sie, daß sich vor dem Starten die Datensätze im Speicher befinden müssen.

IRQ $9D00 – Basisadresse = $9D00 /40192
Speicherbedarf: $9D00 - $9FFF

Auch diese Version ist für eigenständige Intros oder zur Einbindung in Maschinenprogramme gedacht. Geladen wird von der beiliegenden Diskette mit:

```
LOAD"IRQ - $9D00",8,1
```

Beachten Sie, daß sich vor dem Starten die Datensätze im Speicher befinden müssen.

IRQ-Basic – Basisadresse = $0900 /2304
Speicherbedarf: $0801 - $0BFF

Diese Interpreterversion setzt beim Initiieren den Basic-Start hoch, um ein nachfolgendes Basic-Programm zu starten. Damit lassen sich IRQs auch in Basic-Programmen einbinden.

Mit

```
POKE 44,8 : POKE 43,1:SAVE"name",8
```

läßt sich das Basic-Programm zusammen mit dem High-Speed-Interpreter abspeichern.

Mit

```
SYS Basisadresse,Liststart
```

läßt sich eine IRQ-Liste (Befehlsliste) für den High-Speed-Interpreter aktivieren.

Basisadresse – muß als Dezimalzahl eingegeben werden. Für die einzelnen Programmversionen siehe oben.

Liststart – ist die Startadresse der editierten Liste. Sie wird bei jedem Raster-IRQ bearbeitet.

Durch Eingabe von

```
SYS Basisadresse+3
```

wird wieder auf den BASIC-V2 IRQ geschaltet.

2. Der Editor

Mit seiner Hilfe lassen sich die IRQ-Listen editieren, die später vom High-Speed-Interpreter abgearbeitet werden. Geladen wird er mit:

```
LOAD"IRQ DESIGNER",8
```

und gestartet mit RUN. Nach einer kurzen Wartezeit baut sich ein in fünf Bereiche gegliederter Bildschirm auf (Abb. 1):

MAIN-MENÜ

Nach dem Neustart befindet sich der Cursor in diesem Bildschirmbereich. Von hier aus erreichen Sie mit <CURSOR-aufwärts>, <CURSOR-abwärts> und <RETURN> oder den jeweiligen Funktionstasten die acht Menüpunkte des Editors.

Achtung: <RUN/STOP> führt aus allen Menüs zurück ins Hauptmenü.

EDIT-LIST – <F1>

Nach Anwahl dieses Menü-Punkts befindet sich der Cursor im IRQ-LIST Window. Die IRQ-Liste kann editiert werden. Die Befehle finden Sie unter »Befehle des Editors«.

Tastenfunktionen:

<CURSOR aufwärts> – scrollt die Liste nach oben
<CURSOR abwärts> – scrollt die Liste nach unten
<DEL> – löscht aktuellen Befehl
<SHIFT DEL> – fügt einen Befehl ein
<HOME> – an den Listenanfang springen
<RETURN> – Parameter-Eingabe (siehe dort)
<RUN/STOP> – Rückkehr ins Main-Menü

Durch Druck einer der Tasten <A> - <Z> befinden Sie sich im EINGABE-Modus. Der Tastendruck wird als erster Buchstabe übernommen. Im Eingabe-Mode werden die Befehle eingetragen (siehe »Befehle des Editors«). Der Editor

Kurzinfo: IRQ-Designer

Programmart: erzeugt Listen zur Interrupt-Steuerung
Laden: LOAD"IRQ DESIGNER",8
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur
Benötigte Blocks: 73
Programmautor: Markus Hillebrand

Kurzinfo: IRQ-Basic

Programmart: steuert Rasterzeilen-Interrupt
Laden: LOAD"IRQ - BASIC",8
Starten: nach dem Laden RUN eingeben
Steuerung: automatisch über vorher erstellte Liste
Besonderheiten: setzt den Basic-Anfang hoch und startet ein nachfolgendes Programm
Benötigte Blocks: 5
Programmautor: Markus Hillebrand

zeigt manche Befehle in einer ausführlicheren Form an. Sie dürfen aber nur in der unten beschriebenen Form eingegeben werden.

Dazu zwei Beispiele:

Nach Eingabe von »UP« und Bestätigung mit <RETURN> zeigt der Editor »SCR. UP«.

»INC« wird in »INC (+1)« gewandelt.

Fehler bei der Eingabe werden im Feld »REQUESTS« als »ERROR« angezeigt.

Nachdem der Befehl eingegeben ist und der Editor diesen als richtig erkannt hat, befindet sich der Cursor im Parameter-Window. Am linken Window-Rand werden die möglichen Parameter angezeigt. In der Mitte erscheint der hexadezimale und rechts der dezimale Wert. Die Zahl der angezeigten Stellen entscheidet darüber, ob es sich beim Parameter um ein Byte oder ein Wort handelt.

Mit den Cursor-Tasten kann man in diesem Window herumfahren, und beim Drücken einer Ziffer bzw. einer Taste von A-F wird ein neuer Wert eingegeben. Mit <RETURN> kommen Sie ins IRQ-List-Menü zurück.

Der Editor merkt sich, an welcher Position der Cursor im Parameter-Menü stand. War er bei der Rückkehr ins Main-Menü auf einer hexadezimalen Zahl, werden alle Angaben (LOAD/SAVE/NEW/LIST usw.) in HEX abgefragt. Stand er auf einer dezimalen Zahl, wird alles in dezimal erwartet.

NEW LIST – <F2>

Es erscheint im Window »REQUESTS« die Frage nach der Ablageadresse:

```
PLEASE ENTER NEW LOCATION:
....... in dez
```
oder
```
PLEASE ENTER NEW LOCATION:
 $.... IN HEX
```

Geben Sie diese wie gefordert in hexadezimaler oder in dezimaler Schreibweise ein und bestätigen Sie mit <RETURN>.

TEST-LIST – <F3>

Ist dieses Menü angewählt, wird der Bildschirm gelöscht. Die Auswahl zwischen »Input-Pointer« und »List-Pointer« erscheint. Angewählt wird mit dem revers dargestellten Buchstaben.

Input-Pointer: Die Eingabe der Startposition wird erwartet.

List-Pointer: entspricht der aktuellen Position im IRQ-List-Fenster.

Die Frage läßt sich mit <RUN/STOP> abbrechen.

Nach Eingabe des entsprechenden Buchstabens kann frei auf dem Bildschirm editiert werden. <RETURN> führt zurück ins Hauptmenü.

Achtung: Flimmert der IRQ unerträglich, überlagern sich einzelne Befehle. Folgt z.B. ein WAITLINE-Befehl, während der Rasterstrahl diese Position schon lange passiert hat, wartet der HIGH-Speed-Interpreter ein ganzes Bild länger, und die anderen Befehle werden vernachlässigt. Einfach den betreffenden WAITLINE finden und höherstellen.

LOAD LIST – <F4>

Der Menüpunkt heißt zwar »load list«, aber es handelt sich hier um einen universalen LOAD-Befehl. Mit ihm wird es möglich, Programme (damit auch Listen) an beliebige Positionen im Speicher zu laden. Nach Eingabe der Startadresse und des Namens wird das entsprechende File von Diskette an die angegebene Startadresse geladen.

Achtung:

Im Bereich $0801-$4FFF liegt der Editor selbst, also nichts in diesen Bereich laden!

SAVE LIST – <F5>

Genauso universell wie »LOAD LIST«. Nach Eingabe der Start-, Endadresse und des Namens wird das entsprechende File auf Diskette gespeichert.

SHOW DIR – <F6>

Zeigt das Directory der eingelegten Diskette an. Da Floppyfehler nicht abgefragt werden, sollten Sie darauf achten, daß eine Disk eingelegt ist.

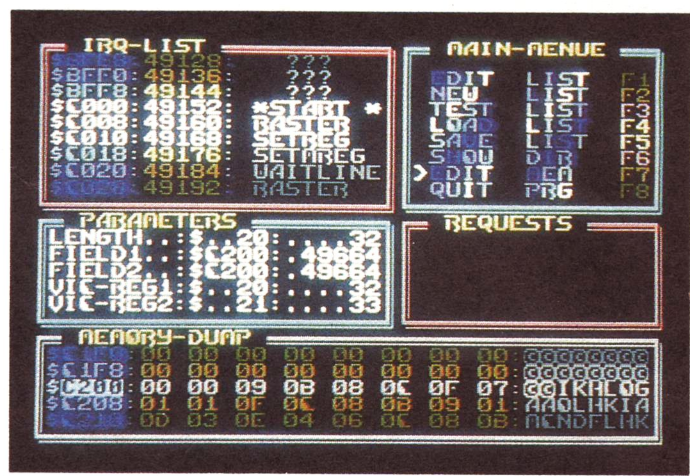


Bild 1. Der IRQ-Designer begrüßt Sie mit dem Hauptmenü

EDIT MEM – <F7>

Aktiviert das Memory-Dump Window. In diesem Window lassen sich die einzelnen FIELDS editieren.

Tastaturbelegung:

<CURSOR aufwärts> – Cursor hochfahren
<CURSOR abwärts> – Cursor nach unten
<RETURN> – ebenfalls Cursor nach unten
<DEL> – ein Byte eliminieren
<SHIFT DEL> – ein 00-Byte einfügen

Soll ein Byte verändert werden, fährt man einfach mit dem Cursor darüber und schreibt einen neuen Wert ein. Wird die Anzeige der Speicheradressen überschrieben, gilt der neue Speicherbereich als angewählt. Auch hier ist $0801-$4FFF tabu. Mit <RUN/STOP> erreichen Sie wieder das Hauptmenü.

QUIT PROGRAMM – <F8>

Beendet nach einer Sicherheitsabfrage im Feld »REQUESTS« den Editor.

Befehle des Editors

Sie werden im Editor in die IRQ-List geschrieben, die Werte für die einzelnen Parameter sind ins Fenster »Parameters« einzutragen. Auf der beiliegenden Diskette befindet sich der Demodatensatz »IRQLIST«. Um die einzelnen Befehle auszuprobieren, empfiehlt es sich, ihn im Editor nach Adresse $C000 zu laden.

RASTER

Mit dem Rasterbefehl ist es möglich, ab der aktuellen Rasterzeile eine bestimmte Anzahl von Daten in zwei VIC-Register zu schreiben, wobei der Schreibvorgang so abgestimmt ist, daß die beiden Register pro Rasterzeile nur einmal beschrieben werden.

Parameter:

LENGTH – Höhe des Rasters in Rasterzeilen
FIELD1 – Speicherbereich der Daten für Register1
FIELD2 – Speicherbereich der Daten für Register2

Achtung:

Die Daten werden beim Rasterbefehl rückwärts gelesen.

VIC-REG1 – Das zu beschreibende Register1
VIC-REG2 – Das zu beschreibende Register2

Achtung:

Da das Timing für die Rahmenfarbe abgestimmt ist, sollte für »REG1« immer Register $D020 verwendet werden. Bei beiden Registern wird nur der Offset angegeben ($20 für $D020).

Ein Beispiel:

```
LENGHT..:$..20:....32
FIELD1..:$C200:.49664
FIELD2..:$C200:.49664
VIC-REG1:$..20:    32
VIC-REG2:$..21:    33
```

Es wird ein Raster mit der Höhe von 32 Rasterzeilen dargestellt, wobei die Daten von 49664 nach $D020 und nach $D021 geschrieben werden.

SETREG

Hier werden zwei Register in der aktuellen Rasterzeile mit den Werten byte1,byte2 beschrieben. Für »REG1« sollte wieder das Register $D020 benutzt werden.

VIC-REG1 - erstes zu änderndes Register
VIC-REG2 - zweites zu änderndes Register
BYTE1 - Wert für Register1
BYTE2 - Wert für Register2

Beispiel:
```
VIC-REG1:$..20:....32
VIC-REG2:$..21:....33
BYTE1...:$..00:.....0
BYTE2...:$..00:.....0
```

Setzt in der aktuellen Rasterzeile die Register $D020/$D021 auf schwarz.

WAITLINE

läßt den High-Speed-Interpreter auf eine bestimmte Rasterzeile warten.

LINE - Nummer der Rasterzeile, auf die gewartet wird.

Beispiel:
```
LINE....:$..64:...100
```

Wartet auf Rasterzeile 100.

Achtung:

Ist die Rasterzeile schon überschritten, wird ein Bildwechsel durchgeführt. Das kann zu Bildschirmflackern führen.

SETMREG

Beschreibt mehrere VIC-Register mit den Daten aus »FIELD« (SET more REGister).

NUMBER - Anzahl der zu kopierenden Daten
FIELD - Adresse, ab der die Daten stehen (aufwärts)
STARTREG - Adresse des ersten Registers

Beispiel:
```
NUMBER..:$..0F:....15
FIELD...:$C230:.49712
STARTREG:$..00:.....0
```

Beschreibt die Register $D000 bis D00F mit Werten aus 49712 bis 49726.

RELWAIT

Läßt den High-Speed-Interpreter eine bestimmte Anzahl von Rasterzeilen warten.

LINES - Anzahl der Rasterzeilen

Beispiel:
```
LINES...:$..0A:....10
```

Wartet zehn Rasterzeilen ab.

INC

Erhöht den Buffer um den Wert 1, verknüpft ihn mit der AND-Maske und der OR-Maske und schreibt den so gewonnenen Wert in ein VIC-Register.

BUFFER - Speicherbereich, dessen Wert um 1 erhöht und anschließend verwendet wird.

Kurzinfo: IRQ-$CD00 und IRQ-9D00

Programmart: steuert Rasterzeilen-Interrupt
Laden: LOAD"IRQ - $CD00",8,1
Laden: LOAD"IRQ - $9D00",8,1
Starten: siehe Anleitung
Steuerung: automatisch über vorher erstellte Liste
Benötigte Blocks: je 3
Programmautor: Markus Hillebrand

Kurzinfo: Example

Programmart: Demoprogramm für IRQ-Designer
Laden: LOAD"EXAMPLE",8
Starten: nach dem Laden RUN eingeben
Steuerung: selbstablaufend
Besonderheiten: lädt den Programmteil »IRQLIST« nach
Benötigte Blocks: 11
Programmautor: Markus Hillebrand

AND-MASK - Wert für die AND-Verknüpfung
OR-MASK - Wert für die OR-Verknüpfung
VIC-REG - zu änderndes Register

Beispiel:
```
BUFFER..:$C000:.49152
AND-MASK:$..0F:....15
OR-MASK.:$..F0:...240
VIC-REG.:$..16:....22
```

Erhöht $C000 um 1, nimmt dann diesen Wert, löscht davon die oberen 4 Bit (or $f0). Bewirkt im Endeffekt, daß das VIC-Register nur in den unteren 4 Bit hochgezählt wird, während die oberen 4 Bit immer gesetzt sind.

DEC

Entspricht dem INC-Befehl, nur wird der Buffer um 1 erniedrigt (siehe »INC«).

UP

Scrollt ab Speicheradresse »FIELD« eine Anzahl Bytes mit variabler Geschwindigkeit nach oben.

SPEED - gibt die Geschwindigkeit an. »0« ist dabei die höchste Geschwindigkeit, »255« ist am langsamsten.
LENGTH - Anzahl der Bytes
FIELD - Speicherstelle, ab der gescrollt wird.

Beispiel:
```
SPEED...:$..01:.....1
LENGTH..:$..0A:....10
FIELD...:$C000:.49152
```

Scrollt 10 Byte ab 49152 in der Geschwindigkeit »1« nach oben.

DOWN

Entspricht »UP«, scrollt aber nach unten.

JSR

Springt in eine Subroutine. Die Subroutine kann sich (muß nicht) dabei die User-Parameter holen.

MEMORY - Adresse der anzuspringenden Routine
USERWORD - Wert zwischen $0000 und $FFFF
USERBYTE - Wert zwischen $00 und $FF

Für die eigene Routine stehen damit 6 Byte zur Verfügung. Sie können folgendermaßen verwendet werden:

```
INY:LDA ($FB),Y ; holt Byte 1
INY:LDA ($FB),Y ; holt Byte 2
```

usw. Der Rücksprung aus dem eigenen Programm erfolgt mit »RTS«.

ENDJMP

LINE - Setzt die Endadresse, die der High-Speed-Interpreter nach dem Auffinden eines STOP-Befehls anspringt.

Achtung:

Wird erst nach einem STOP-Befehl angesprungen. Ohne diesen wird automatisch nach $EA31 gesprungen.

START

LINE - gibt die Rasterzeile an, in der der HIGH-Speed-Interpreter zu arbeiten beginnt.

STOP

Beendet den Raster-Interrupt und springt (wenn kein END-JUMP angegeben wurde) nach $EA31!

Übrigens! IRQ-Listen können frei im Speicher verschoben werden, da sie keine JMPs oder JSRs besitzen.

Der High-Speed-Interpreter versucht zwar, die Befehle möglichst flimmerfrei auf den Bildschirm zu bringen, aber es geht leider nicht immer 100prozentig. Fängt z.B. ein Rasterbefehl in einer »magischen« 8-Zeile an, flimmert er. In diesem Falle erhöhen Sie einfach den Waitbefehl davor um eins oder erniedrigen ihn. Das wirkt Wunder.

Auf der beiliegenden Diskette befindet sich ein Demoprogramm. Sie laden es mit:

```
LOAD"EXAMPLE",8
```

und starten es mit RUN. Nachdem der Programmteil »IRQ-LIST« nachgeladen ist, zeigt es Ihnen einige Möglichkeiten des »IRQ-Designers«. Unterbrechen Sie doch mal mit <RUN/STOP RESTORE> und listen Sie den Basic-Teil mit »LIST«. Sie sehen, wie man mit wenig Aufwand einen Titel gestalten kann.

(Martin Schweiger/gr)



Magazin im Überblick

Diese 64´er-Ausgaben bekommen Sie noch bei Markt & Technik für jeweils 6,50 DM, ab der Ausgabe 1/90 für 7,- DM, der Preis für Sonderhefte und Sammelboxem beträgt je 16,- DM. Tragen Sie Ihre Bestellung im Bestellcoupon ein und schicken Sie ihn am besten gleich los oder rufen Sie einfach an.

10/89: Listing des Monats: Power-Musik-Editor / Test Handyscanner / 64ér-Longplay: Grant Monster Slam

11/89: Super-Drucker uner 600 Mark / Der Zeichen-Künstler MonoöMagic / Grafikduell C64, Amiga Atari ST, PC

1/90: Gratis: BTX für alle! Mit Diskette im Heft / Joysticktest / Heimcomputer im DFÜ-Vergleich / Hurrican - die neue Spiele-Dimension

2/90: Systemvergleich: Die besten BTX-Dexoder / Funken mit dem C64 / Musik: "Power DIGI Editor" / 64er-Longplay "Oil Imperium"

3/90: Neue Speichertechniken / Grafikduell mit dem PC, Atari ST, Amiga und C64 / Neue Referenz: Brother M1826 I

4/90: Die Geos-Welt: das komplette Geos-System; Geos-Poster / Test Videofoy / Programm des Monats: Topprint

05/90: Listings des Monats: Sternwelt / Bauanleitung: Regelbares Dauerfeuer / Test Spielpack: Top oder Flop

6/90: Programmierung: endlich Basic 3.5 für C64 / Softwaretest: die besten Fußballprogramme / Videostudio, C64 in Börsenfieber

7/90: Extratouren: CD-Musicbox mit C64 und Bauanleitung Pulsmesser / Sammelposter C64 im Riesenforma

9/90: Großer C64-Reparaturkurs / Faszination: Amateurfunk / Neuigkeiten aus der Geos-Welt / Super-Spile zum Abtippen

10/90: Bauanleitungen: 5 Wochenend-Projekte / ECOM-das Super-Basic / Test: Die besten Drucker unter 1000 DM / C64-Reparaturkurs

11/90: Bausatztest: Der Taschengeldplotter / Vergleichstest: Drucker der Spitzenklasse / 5 Schnellbauschaltungen

12/90: Abenteuer BTX / Multitasking für C64 / Großer Spieleschwerpunkt / Programmierwettbewerb: 30 000 DM zu gewinnen

01/91: Die Besten Tips&Tricks / Neu: Reparaturecke / Floppy-Flop: Betriebssystem überlistet / Jahresinhaltsverzeichnis

2/91: Sensation: Festplatte für den C64 / Alles rund um den Drucker / Longplay: Dragon-Wars / Neu: Spiele-Tips

Mit diesen Sammelboxen sind Ihre Ausgaben immer sortiert und griffbereit



Eine Sammelbox faßt einen vollständigen Jahrgang mit 12 Ausgaben und kostet 14,- DM. Bestellen Sie sie mit nebenstehendem Coupon.

Ab sofort können Sie auch telefonisch bestellen unter 089/20251527



Sonderhefte im Überblick

Die 64´er Sonderhefte bieten Ihnen umfassende Information in komprimierter Form zu speziellen Themen rund um die Commodore C 64 und C 128. Ausgaben, die eine Diskette enthalten, sind mit einem Diskettensymbol gekennzeichnet,

GRAFIK, ANWENDUNGEN, SOUND



SH 0020: Grafik
Grafik-Programmierung / Bewegungen



SH 0031: DFÜ, Musik, Messen-Steuern-Regeln
Alles über DFÜ / BTX von A-Z / Grundlagen / Bauanleitungen



SH 0045: Grafik
Listings mit Pfiff / Alles über Grafik-Programmierung / Erweiterungen für Amica-Paint



SH 0046: Anwendungen
Das erste Expertensystem für den C 64 / Bessere Noten in Chemie / Komfortable Dateiverwaltung



SH 0053: Das Beste aus 5 Jahren
10 Top-Programme aus allen Bereichen / PC-Simulationen aus dem C64



SH 0055: Grafik
Amica-Paint: Malen wie ein Profi / DTP-Seiten vom C64 / Tricks&Utilities zur Hires-Grafik

PROGRAMMIERSPRACHEN



SH 0056: Anwendungen
Gewinnauswertung beim Systemlotto / Energie-verbrauch voll im Griff / Höhere Mathematik und C64



SH 0035: Assembler
Abgeschlossene Kurse für Anfänger und Fortgeschrittene



SH 0040: Basic
Basic Schritt für Schritt / Keine Chance für Fehler / Profi-Tools und viele Tips

FLOPPYLAUFWERKE, DATASETTE, DRUCKER



SH 0025: Floppylaufwerke
Wertvolle Tips und Informationen für Einsteiger und Fortgeschrittene



SH 0032: Floppylaufwerke und Drucker
Tips&Tools / RAM-Erweiterung des C64 / Druckerroutinen



SH 0047: Drucker, Tools
Hardcopies ohne Geheimnisse / Farbige Grafiken auf s/w-Druckern

C 64, C 128, EINSTEIGER



SH 0022: C 128 III
Farbiges Scrolling im 80-Zeichen Modus / 8-Sekunden-Kopierprogramm



SH 0026: Rund um den C64
Der C64 verständlich für Alle mit ausführlichen Kursen



SH 0029: C 128
Starke Software für C 128/ C 128D / Alles über den neuen C 128D im Blechgehäuse



SH 0036: C 128
Power 128: Directory komfortabel organisieren / Haushaltsbuch: Finanzen im Griff / 3D-Landschaften auf dem Computer



SH 0038: Einsteiger
Alles für den leichten Einstieg / Super Malprogramm / Tolles Spiel zum selbermachen / Mehr Spaß am Lernen



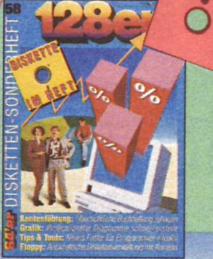
SH 0044: C 128
Grafikspeicher auf 64KB erweitern / Leistungstest GEOS 128 2.0 / Tips zum C 128



SH 0050: Starthilfe
Alles für den leichten Einstieg / Heiße Rythmen mit dem C 64 / Fantastisches Malprogramm



SH 0051: C 128
Volle Floppy-Power mit "Rubikon" / Aktienverwaltung mit "Börse 128"



SH 0058: 128er
Übersichtliche Buchhaltung zuhause / Professionelle Diagramme

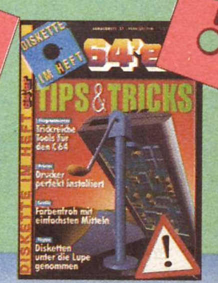
TIPS, TRICKS & TOOLS





SH 0024: Tips, Tricks& Tools
Die besten Peeks und Pokes sowie Utilities mit Pfiff

SH 0043: Tips, Tricks&Tools
Rasterinterrupts - nicht nur für Profis / Checksummer V3 und MSE / Programmierhilfen



SH 0057: Tips & Tricks
Trickreiche Tools für den C64 / Drucker perfekt installiert

DTP

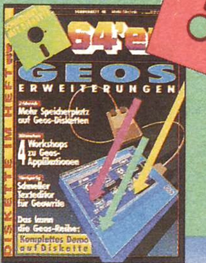


SH 0039: DTP, Textverarbeitung
Komplettes DTP-Paket zum Abtippen / Super Textsystem / Hochauflösendes Zeichenprogramm

GEOS, DATEIVERWALTUNG



SH 0028: Geos / Dateiverwaltung
Viele Kurse zu Geos / Tolle Geos-Programme zum Abtippen



SH 0048: GEOS
Mehr Speicherplatz auf Geos-Disketten / Schneller Texteditor für Geowrite / Komplettes Demo auf Diskette



SH 0059: GEOS
GeoBasic: Großer Programmierkurs mit vielen Tips & Tricks

SPIELE



SH 0030: Spiele für C 64 und C 128
Tolle Spiele zum Abtippen für C 64/C 128 / Spieleprogrammierung



SH 0037: Spiele
Adventure, Action, Geschicklichkeit / Profihilfen für Spiele / Überblick und Tips zum Spielekauf

SH 0042: Spiele
Profispiele selbst gemacht / Adventure, Action, Strategie



SH 0049: Spiele
Action, Adventure, Strategie / Sprites selbst erstellen / Virenkiller gegen verseuchte Disketten



SH 0052: Abenteuerspiele
Selbstprogrammieren: Von der Idee zum fertigen Spiel / So knacken Sie Adventures



SH 0054:
Adventures, Science Fiction, Horror / Viel Spaß mit "Rubberball"



SH 0060: Adventures
8 Reisen ins Land der Fantasie - so macht Spannung Spaß



SH 0061: Spiele
20 Heiße Super Games auf Diskette

BESTELLCOUPON

Ich bestelle die 64er Sonderhefte Nr. ____________

zum Preis von je
14,- DM (Heft ohne Diskette), 16,- DM (Heft mit Diskette)
24,- DM (nur für die Ausgabe SH 0051 + 0058)
Ich bestelle das 64er Magazin Nr. ____________

zum Preis von je
6,50 DM (bis Ausgabe 12/89), 7,- DM (ab Ausgabe 1/90)
.... Sammelbox (en) zum Preis von je 14,- DM
zzgl. Versandkosten
Ich bezahle den Betrag nach Erhalt der Rechnung

Name, Vorname

Straße, Hausnummer

PLZ, Wohnort

Schicken Sie bitte den ausgefüllten Bestellcoupon an:
Markt&Technik Leserservice, CSJ, Postfach 140 220,
8000 München 5, Tel.: 089/ 20 25 15 27

Durch fantastische Intros zeichnen sich fast alle professionellen Programme auf dem C64 aus. Jetzt können Sie diese auch selbst machen.

Star-Demomaker

Neue Dimensionen

B. Bommer

Der »Star-Demomaker« ist vollständig menügesteuert. Durch die Tasten <1>-<7> lassen sich in den verschiedenen Menüs und Untermenüs die einzelnen Punkte anwählen. Sie laden das Programm von der beiliegenden Diskette mit

```
LOAD"DEMOMAKER (15)",8
```

und starten mit RUN. Anschließend sehen Sie das Titelbild mit den einzelnen Menüs.

<1> - Watch Demo

zeigt das Demo. Im Demo kann die Geschwindigkeit des Scrollers mit einem Joystick in Port 2 beeinflußt werden.

<RESTORE> führt ins Hauptmenü zurück.

<2> - Editmen

dient der Bearbeitung und Neueinrichtung. Alle Änderungen sind sofort aktiv und können mit »Watch Demo« begutachtet werden.

Im Editmenü stehen fünf Untermenüs zur Verfügung:

<1> - Change Colour of Rasters

Hier lassen sich Farben der Raster und des Scrollers einstellen.

<+> erhöht den Farbwert (0-15),

<-> vermindert ihn.

<SPACE> setzt den Cursor ein Feld weiter nach links,

<RETURN> führt eine Zeile weiter.

<G> schaltet als besonderes Bonbon den »Genlock-Mode« ein oder aus. Das macht den Demomaker zum leicht zu bedienenden Genlock-Titelgenerator.

<2> - Raster through

stellt die Position des Rasterbalkens ein.

<Both> bewegt durch Border und Bildschirm.

<Border> bewegt nur durch den Bildrahmen,

<Screen> nur durch den Bildschirm.

<3> - Musikparameter

Mit jedem Tastendruck wird einer der Musiktypen eingestellt und angezeigt:

»Music $1000/$1006« ist eine Musik mit INIT $1000 und IRQ - $1006.

»Music $1000/$1003« entspricht »$1000/$1006«, aber der IRQ-Einsprung liegt bei $1003.

<Grumbling> erzeugt einen Brummton,

<Silence> - Stille.

<4> - Edit Text

führt in den Scrolltexteditor. Alle Buchstaben, auch »!«, »?«, »,« und ».« können eingegeben werden. Ebenso sind Cursor-Tasten und die INS/DEL-Taste aktiv.

<SHIFT RETURN> verläßt den Editor.

Achtung: Der Editor benutzt einige Pointer in der Zeropage. Einige Musikstücke überschreiben diese. Sollten Sie ein seltsames Verhalten bei der INS/DEL-Funktion feststellen, fahren Sie mit der <CRSR runter>-Taste an das Textende und editieren dann wie gewohnt.

<5> - Go Back to Main Menu

Führt zurück ins Hauptmenü.

<3> - Diskmen

Hier werden alle Floppyoperationen durchgeführt.

<1> - Load Music

Nach Eingabe des File-Namens wird der Datensatz an die Speicherposition $1000 geladen. Danach sollten Sie im Editmenü die Musikparameter einstellen.

<2> - Load Wave

Eine neue Sinuswelle für den Rasterbalken wird geladen.

<3> - Load Char

lädt einen neuen Sprite-Zeichensatz.

<4> - Load Picture

lädt Bilder im Koala-Format. Der Name muß mit Prefix (?pic ...*) eingegeben werden. Selbstverständlich kann auch ein im Koala-Format gespeichertes Amica-Paint-Bild geladen werden.

<5> - Save whole Demo

Ein fertiges Demo wird unter dem eingegebenen Namen gespeichert. Wenn möglich, sollte man das Demo danach crunchen (packen). Die Startadresse ist $5003 (hex.)/20483 (dez.). Das Demo läßt sich ungecruncht mit RUN starten.

<6> - Send Disccommand

Ein einzugebender String von max. 40 Zeichen Länge wird an die Floppy geschickt. Bei Eingabe von »$« und <RETURN> wird das Directory gelistet. Die Ausgabe wird mit Space gestoppt und mit <RUN/STOP> abgebrochen.

<7> - Back to Main Menu

Führt zurück ins Hauptmenü.

Tips zur Benutzung anderer Musik

Die Musik darf zwischen $1000 und $2000 liegen. Der INIT-Einsprung muß bei $1000, der IRQ-Einsprung bei $1003 oder $1006 sein. Die am häufigsten benutzten Editoren sind der Future Composer und Romuzak. Beide besitzen eine Relocate-Funktion (beim Future Composer ab Version 2.1). Mittels dieser Funktion muß die Musik nach $1000 verschoben werden. Im Editmenü stellen Sie dann den Einsprung ein: beim Future Composer $1000/$1006, bei Romuzak $1000/$1003.

Sollten Sie andere Musik verwenden, überprüfen Sie, ob diese bei $1000 startet. Ist dies der Fall, probieren Sie beide Einstellungen ($1003/6) aus. Bei einem eventuellen Absturz laden Sie den Editor einfach neu.

<4> - Reset

Nach einer Sicherheitsabfrage wird der Editor durch ein Reset verlassen.

Tips zur Erstellung eigener Zeichensätze

Die Zeichensätze können mit jedem normalen Sprite-Editor erstellt werden. Das Alphabet und die Sonderzeichen »!,.?« sind zu zeichnen. Speichern Sie die Sprites ab. Der Demomaker lädt sie automatisch nach $0840 bis $1000. Von $0840 bis $0EC0 liegen die Zeichen des Alphabets. Von $0EC0-$0F00 liegt das Komma, von $0F00-$0F40 der Punkt, von $0F40-$0F80 das Ausrufungszeichen und von $0F80-$0FC0 das Fragezeichen. Der Bereich von $0FC0-$1000 steht für Space und muß deshalb leer sein.

Kurzinfo: Star-Demomaker

Programmart: Intromaker
Laden: LOAD"DEMOMAKER (15)",8
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur und Joystick in Port 2
Benötigte Blocks: 31
Programmautor: B. Bommer

Movement-Editor



wie im Film

Sie faszinieren immer wieder, die tollen Sprite-Bewegungen in Spielen oder bei Intros. Mit dem Movement-Editor holen auch Sie solche grafischen Spitzenleistungen aus dem C64 heraus. Leisten Sie sich Ihr eigenes Sprite-Ballett.

A. Kunze

Erstaunt betrachtet man bei vielen professionellen Programmen, wie eine Kette von Sprites im Formationsflug über den Bildschirm huscht (Abb. 1). Dazu sind keine tiefgehenden Assembler-Kenntnisse nötig: Der Movement-Editor nimmt Ihnen die größte Arbeit ab, nämlich die Bewegungszüge im Interrupt abzufahren. Die nötigen Sprites konstruieren Sie mit dem im Heft beschriebenen Sprite-Editor (siehe S. 44). Es bleibt nur noch, die VIC-Register geeignet zu setzen und die vom Movement-Editor erzeugte Abspielroutine zu aktivieren. Doch zuerst das Funktionsprinzip:

Eine Kette, gebildet aus bis zu acht Sprites, soll über den Bildschirm fahren. Dazu könnte man einfach alle x- und y-Positionen speichern. Im Anschluß daran könnten alle Sprites hintereinander diese Positionskette abfahren. Aber wie sieht es mit dem Speicherplatzverbrauch aus? Es sind 3 Byte nötig – zwei für die x-Koordinate, und eines für die y-Position. Bei dem Movement-Editor haben wir uns für eine andere Methode entschieden – die relative Datenspeicherung:

Ein Byte hat bekanntlich ein Hi- und Lo-Nibble, diese haben wiederum 4 Bit. Setzt man nun für jede Bewegung in eine bestimmte Richtung 1 Bit, kommen auf 1 Byte zwei Bewegungsschritte, für die ansonsten 6 Byte benötigt würden. Also ein merkbar geringerer Speicherbedarf.

Um acht Sprites im korrekten Abstand zueinander zu bewegen, muß der Movement-Editor eine Tabelle aufbauen, in der die darin enthaltenen Speicherpositionen hochgezählt werden. Man kann sich das wie eine Schnur vorstellen, an der sich jedes Sprite entlanghangelt.

Problematisch ist alleine der Start der Sprites, da eines nach dem anderen starten soll. Dafür existiert extra eine Vorspur.

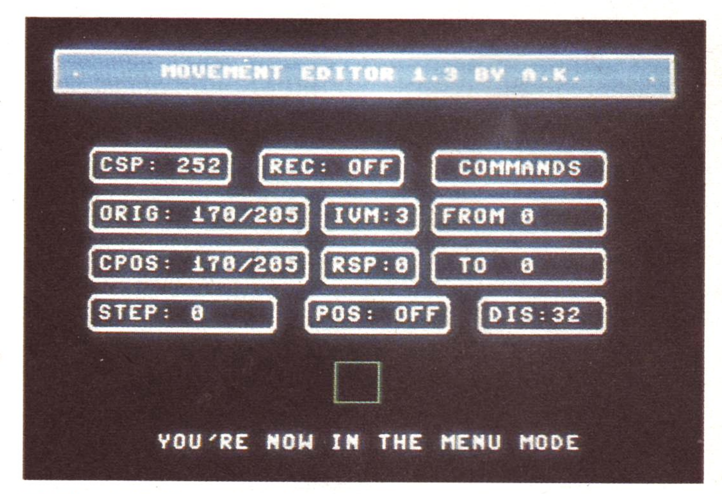
Sie laden das Programm durch Eingabe von:

```
LOAD"MOVEMENTEDITOR",8
```

und starten mit RUN. Danach sehen Sie die Eingabemaske und unterhalb der Bildschirmmitte ein quadratisches Cursorsprite. Aus der untersten Bildschirmzeile, der Kommandozeile, ersehen Sie, in welchem Modus Sie sich befinden. Nach dem Start ist dies der Menümodus. Die einzelnen Bezeichnungen der Fenster am Bildschirm (Abb. 1) haben folgende Bedeutung:

CPOS bezeichnet die Cursor-Position.
ORIG bezeichnet die Home-Position.
DIST bezeichnet die Spritedistanz.
RSP bezeichnet die Aufnahmegeschwindigkeit.
IVM bezeichnet den Invertierungsmodus.
STEP bezeichnet die aufgenommenen Datenbytes.
REC gibt an, ob die Aufnahmeroutine ein- oder ausgeschaltet ist.
POS gibt an, ob die ständige Positionsanzeige ein- oder ausgeschaltet ist.
COMMANDS bezeichnet das derzeitige Edit-Kommando.
FORM/TO bezeichnet die eingegebenen Zahlen.

Zur Bedienung des Movement-Editors existieren zwei Gruppen von Befehlen.



[1] Die vielfältigen Anzeigen des Movement-Editors zeugen von seiner Leistungsfähigkeit

1. Grundeinstellungen

Erreicht werden diese grundsätzlich aus dem Menümodus. Sie müssen vor der Aufnahme der Bewegungsabläufe benutzt werden, da sie die Initiierungstabellen beeinflussen:

<D> – Distanz

in der untersten Bildschirmzeile erscheint der Text »Your Sprite Distance are now 32 steps«. Diese Distanz ist beim späteren Abspielen der Zwischenraum zwischen den Sprites.
<+> oder <-> ändert diese Distanz.

<SPACE> schließt die Eingabe ab und führt zurück in den Menümodus.

<O> – Home

definiert eine neue Home-Position der Sprites, die bei jedem Abspielen der Daten den Startpunkt darstellt. Drücken von <HOME> führt den Cursor aus dem Menümodus immer an diese Position zurück.

2. Editierbefehle

In diese einzelnen Modi kommen Sie über die entsprechende Taste aus dem Menü-Modus. Verlassen werden sie durch <SPACE>.

<T> - Trace
schaltet die Aufzeichnung von Bewegungsdaten in den Speicher an oder aus. Den jeweiligen Zustand sehen Sie im Fenster »REC:«.

<J> - Joystick
aktiviert die Cursor-Steuerung durch den Joystick (Port 2). Die Bahnen, die später die Sprites abfahren sollen, lassen sich jetzt bequem mit dem Joystick festlegen.

<DEL> ergibt eine Positionsanzeige des Cursors im Fenster »CPOS«.

<SPACE> oder ein Druck auf den Feuerknopf beendet diese Funktion.

Achtung:
Bewegungsdaten werden erst aufgezeichnet, wenn <T> aktiviert ist (s. o.).

<K> - Keyboard
Die Funktion ist wie bei »Joystick«, nur geschieht die Steuerung mit Hilfe der Tastatur. Folgende Tasten sind aktiv:

<Klammeraffe> höher
<:> links
<;> rechts
</> unten
<DEL> ergibt eine Positionsanzeige des Cursors im Fenster »CPOS«.

<L> - Linien
erlaubt, Linien in beliebigem Winkel zu ziehen. Gesteuert wird über die Tastatur (s. Keyboard). Der alte Endpunkt wird der neue Startpunkt.

<RETURN> läßt den Cursor die Linie abfahren.

<DEL> ergibt eine Positionsanzeige des Cursors im Fenster »CPOS«.

<S> - Speed
<-> verringert die Geschwindigkeit
<+> erhöht die Geschwindigkeit
<*> stellt zurück auf Standardwert

<P> - Play
stellt eine Kontrollfunktion für die Aufzeichnung dar. Der Cursor fährt die eingegebene Bahn ab.

<Pfund> - Replay mit acht Sprites
Fährt nacheinander die Bahnen der Sprites ab. Die immer von neuem beginnende Routine wird mit <SPACE> unterbrochen.

<E> - Editieren
Haben Sie vielleicht einen Fehler bei der Eingabe gemacht? In diesem Menü läßt er sich rückgängig machen. Ebenso sind zusätzliche Funktionen erreichbar:

<D>: Daten löschen. Sie geben einfach den Anfang der zu löschenden Daten ein, und das Programm setzt ein neues Ende der Daten.

<C>: Daten kopieren. Nach der Eingabe der Anfangs- und Endadresse des zu kopierenden Datenteils erfolgt eine Plausibilitätskontrolle.

<I>: eine Spezialität des Programms ist das Invertieren von Bewegungsdaten. Das kann nützlich sein, wenn Sie z.B. eine bestimmte Figur spiegelverkehrt abspielen wollen. Um einzustellen, was getauscht werden soll, drücken Sie, gegebenenfalls mehrfach, die Taste <M>. Im Fenster >IVM< erscheint nun eine >1< (alles vertauschen). Wenn oben mit unten getauscht werden soll, eine >2<, wenn links mit rechts getauscht werden soll, und eine >3<, wenn beides geschehen soll. Haben Sie sich entschieden, drücken Sie <SPACE>, und die Daten werden getauscht.

Achtung:
Leider sind nicht alle C 64 baugleich. Bei einigen Modellen führt der Menüpunkt »Invertieren« zu einem Absturz des Systems.

<DEL> - Positionsausgabe
gibt die derzeitige Position aus (Anzeige im CPOS-Fenster).

<R> - Restart
Stellt alle Daten auf Standardwerte und den Steppointer auf 0. Dieser Tastendruck bewirkt eine Sicherheitsabfrage. Erst wenn diese mit <Y> beantwortet ist, wird der Warmstart durchgeführt.

<Q> - Quit
führt einen RESET des Computers nach einer Sicherheitsabfrage durch.

<F> - Filefunktionen
Um die Daten samt der Abspielroutine in Ihr eigenes Programm einzubinden, drücken Sie <F>. Hier wählen Sie nun zwischen LOAD oder SAVE. Es werden noch folgende Fragen gestellt:

- Filename
- Startadresse (hexadezimal), der Editor rechnet automatisch die Absolutadressen der Bewegungsroutine um. Das heißt, die Routine ist danach nicht mehr frei im Speicher verschiebbar.
- Anzahl der Sprites, die später bewegt werden sollen. Es werden dabei die Sprites von Sprite 0 an aufwärts benutzt.

Wichtige Daten der Abspielroutine:
Die Startadresse ist von Ihnen beim Speichern festgelegt worden. Der Einsprung in die einzelnen Routinen erfolgt mit

```
SYS Adresse
```

Startadresse = Einsprung, um ein Sprite einen Schritt zu bewegen.

Startadresse + 11 = Init, notwendiger Einsprung, um verschiedene Zeiger richtig zu setzen.

Startadresse + 242 und folgende Bytes: x- und y-HOME-Position und x-Hibyte.

Nachdem Sie die Daten gespeichert haben, können Sie sie wieder laden und weiterbearbeiten.

Einbau in eigene Programme
Alle Videoregister, die Sprites betreffen, sind vor dem Einsprung zu setzen.

Die Initialisierung erfolgt bei Startadresse + 11 mit JSR start +11.

Für die Dauer des Abspielvorgangs muß jetzt immer wieder mit JSR start in die Bewegungsroutine eingesprungen werden.

Jedesmal bewegen sich die Sprites um einen Schritt weiter.

Kurzinfo: Movement-Editor

Programmart: Sprite-Beweger für Intros oder Spiele
Laden: LOAD"MOVEMENTEDITOR",8
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur oder Joystick
Benötigte Blocks: 21
Programmautor: A. Kunze

Noch ein paar Tips
Es ist am besten, einen Rasterzeilen-Interrupt im Bereich der Border festzusetzen; dadurch werden Verzerrungen der Sprites ausgeschlossen. Haben Sie aus Versehen eine falsche Funktion gewählt, ist es immer möglich, mit <SPACE> oder <RETURN> zurückzukehren.

Versuchen Sie einmal durch Setzen unterschiedlicher Startpositionen, das heißt Aufteilen der Sprites in zwei Gruppen, verschiedene Effekte zu erreichen.

Wenn Sie wollen, daß die Sprites nicht von einem Punkt aus starten, geben Sie als Distanz 0 ein und setzen von Ihrem aufrufenden Programm aus die Spritepositionen so, daß alle hintereinander auf einer Linie in Startrichtung liegen (Sprite Null als erstes). Um Ihnen einen Vorgeschmack auf die Fähigkeiten des Movement-Editors zu geben, befindet sich auf der beiliegenden Diskette ein Demofile. Sie laden es mit

```
LOAD"MOVEMENT.DEMO",8
```

und starten mit RUN.

(gr)

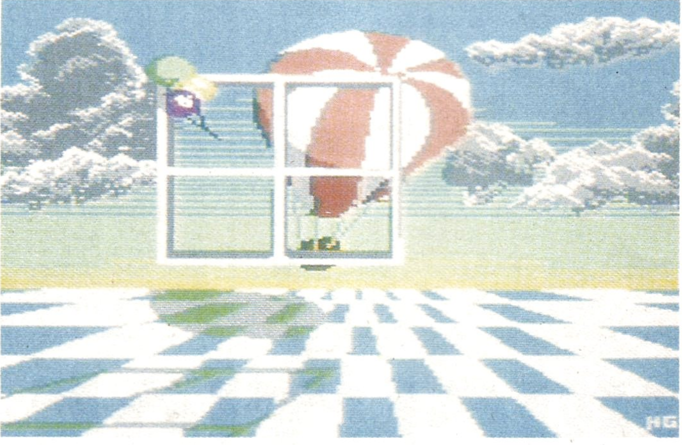
Die besten Colorbilder

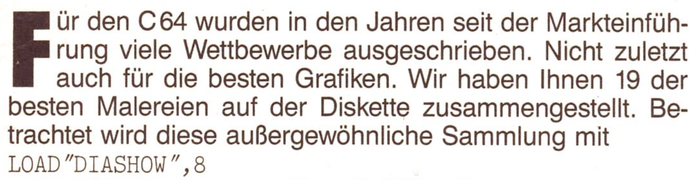
Picture-Show

Perfekte Grafiken sind mit dem C64 nicht möglich – wird jeder Amigabenutzer sagen. Daß es doch geht, zeigt unsere Diashow.









Für den C64 wurden in den Jahren seit der Markteinführung viele Wettbewerbe ausgeschrieben. Nicht zuletzt auch für die besten Grafiken. Wir haben Ihnen 19 der besten Malereien auf der Diskette zusammengestellt. Betrachtet wird diese außergewöhnliche Sammlung mit

```
LOAD"DIASHOW",8
```

Anschließend starten Sie mit RUN. Das Hauptmenü erlaubt Ihnen die Auswahl zwischen:

<**F1**> **– Diashow ablaufen lassen**
führt alle 19 Bilder in Reihenfolge vor. Ist ein Bild geladen, wird das durch Blinken des Bildschirmrahmens angezeigt. <SPACE> bringt dieses Gemälde auf den Schirm.

<**F3**> **– Einzelbild ansehen**
zeigt eines der Bilder. Sie haben die Auswahl zwischen
<F5> Nummer eingeben und
<F7> Name eingeben.
Ist das gewünschte Bild nicht auf Diskette (z.B. »20«), führt <SPACE> zurück ins Hauptmenü.

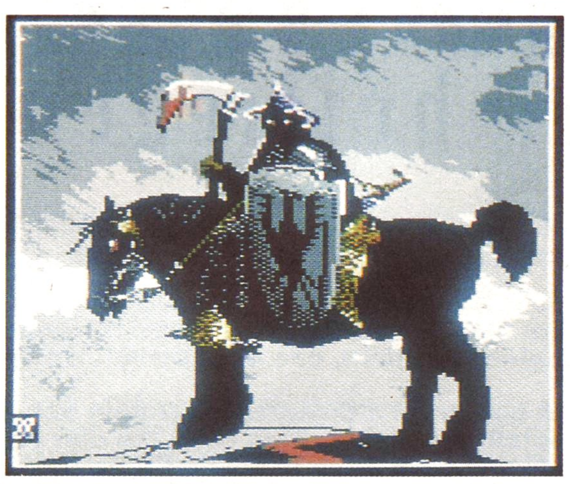
<**F5**> **– Directory anzeigen**
zeigt das Inhaltsverzeichnis der eingelegten Diskette an.

<**F7**> **– Ende**
Beendet die Diashow ohne Sicherheitsabfrage. Ein RESET wird dabei durchgeführt.

Ein Neustart ist mit: SYS 49152 möglich.



(gr)



Kurzinfo: Diashow

Programmart: Grafikdemo
Laden: LOAD"DIASHOW",8
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur
Benötigte Blocks: 5 für das Betrachterprogramm

So finden Sie die Programme auf der Diskette

DISKETTE SEITE 1

```
0  "64'ER SONDERH.63" S1 2A
29    "DISKLADER"            PRG
0     "----------------"     DEL
0     "----------------"     DEL
0     "--MALPROGRAMM---"     DEL
0     "----------------"     DEL
0     "----------------"     DEL
42    "EGA V3.2 C 64"        PRG   Seite 4
42    "EGA V3.2 C128"        PRG
3     "EGA - PRINT"          PRG
27    "ZEICHENSATZ 1"        PRG
8     "ZEICHENSATZ 2"        PRG
31    "ZEICHENSATZ 3"        PRG
4     "EGA-PRINT.GEN"        PRG
4     "EGA - PRINT.GEN+"     PRG
0     "----------------"     DEL
0     "----------------"     DEL
0     "-----INTROS-----"     DEL
0     "----------------"     DEL
0     "----------------"     DEL
73    "IRQ DESIGNER"         PRG   Seite 9
5     "IRQ - BASIC"          PRG
3     "IRQ - $CD00"          PRG
3     "IRQ - $9D00"          PRG
4     "IRQLIST"              PRG
7     "EXAMPLE"              PRG
0     "----------------"     DEL
31    "DEMOMAKER (15)"       PRG   Seite 14
40    "DEMONSTRATION"        PRG
0     "----------------"     DEL
21    "MOVEMENTEDITOR"       PRG   Seite 15
6     "MOVEMENT.DEMO"        PRG
```

```
0     "----------------"     DEL
0     "----------------"     DEL
0     "---ANWENDUNG----"     DEL
0     "----------------"     DEL
0     "----------------"     DEL
4     "GRAPH"                PRG   Seite 29
11    "INIT"                 PRG
80    "HPROGRAMM"            PRG
32    "HMACHER"              PRG
12    "SMACHER"              PRG
8     "STRINGS"              PRG
12    "0HILFE"               PRG
1     "1HILFE"               PRG
1     "2HILFE"               PRG
1     "3HILFE"               PRG
1     "4HILFE"               PRG
2     "5HILFE"               PRG
1     "6HILFE"               PRG
1     "7HILFE"               PRG
1     "8HILFE"               PRG
1     "9HILFE"               PRG
1     "10HILFE"              PRG
1     "11HILFE"              PRG
1     "12HILFE"              PRG
1     "13HILFE"              PRG
1     "14HILFE"              PRG
0     "----------------"     DEL
0     "----------------"     DEL
0     "-TIPS & TRICKS--"     DEL
0     "----------------"     DEL
0     "----------------"     DEL
8     "DELUXE DATAS"         PRG   Seite 22
```

```
0     "----------------"     DEL
59    "B.M.G"                PRG   Seite 28
3     "64'ER.MSK"            PRG
2     "ADRESS.MSK"           PRG
0     "----------------"     DEL
6     "BALKENSTATISTIK"      PRG   Seite 23
0     "----------------"     DEL
1     "SCREEN-TOOL"          PRG   Seite 23
0     "----------------"     DEL
1     "SPRITESET"            PRG   Seite 23
0     "----------------"     DEL
2     "LORES"                PRG   Seite 23
0     "----------------"     DEL
1     "LAUFSCHRIFT"          PRG   Seite 23
3     "BASIC-DEMO"           PRG
0     "----------------"     DEL
6     "ECR - PLOT"           PRG   Seite 23
6     "ECR-DEMO"             PRG
0     "----------------"     DEL
0     "----------------"     DEL
0     "----SPRITES-----"     DEL
0     "----------------"     DEL
0     "----------------"     DEL
6     "DEMO1"                PRG   Seite 47
0     "----------------"     DEL
0     "----DISKETTE----"     DEL
0     "---BEIDSEITIG---"     DEL
0     "----BESPIELT----"     DEL
0     "----------------"     DEL
3 BLOCKS FREE.
```

DISKETTE SEITE 2

```
0  "64'ER SONDERH.63" S2 2A
29    "DISKLADER"            PRG
0     "----------------"     DEL
0     "----------------"     DEL
0     "-----SPRITES----"     DEL
0     "----------------"     DEL
0     "----------------"     DEL
19    "TEGRA"                PRG   Seite 40
11    "HI-TEGRA"             PRG
3     "CONVERTER"            PRG
1     "CONVERTER $9F00"      PRG
9     "WELTKARTE.TEGRA"      PRG
33    "WELTKARTE.HIRES"      PRG
0     "----------------"     DEL
10    "VIEW-INSTALL"         PRG   Seite 44
8     "VIEW BASIC"           PRG
8     "VIEW 49152"           PRG
```

```
5     "SPR.GARFIELD"         PRG
14    "SPR.WIZBALL"          PRG
34    "SPR.BUBBLE"           PRG
0     "----------------"     DEL
0     "----------------"     DEL
0     "----DIASHOW-----"     DEL
0     "----------------"     DEL
0     "----------------"     DEL
5     "DIASHOW"              PRG   Seite 17
26    "▪01 ARC DE TR."       PRG
18    "▪02 MARYLIN"          PRG
15    "▪03 SNOWCAR"          PRG
31    "▪04 TEMPEL"           PRG
14    "▪05 CHIP"             PRG
25    "▪06 FANTASY"          PRG
21    "▪07 EISVOGEL"         PRG
26    "▪08 DINOSAURIER"      PRG
```

```
23    "▪09 DER 12. TAG"      PRG
21    "▪10 DEATH RIDER"      PRG
23    "▪11 RHEA"             PRG
27    "▪12 PHARAO"           PRG
37    "▪13 SURFER"           PRG
20    "▪14 PONT DU GARD"     PRG
25    "▪15 BRUCE LEE"        PRG
33    "▪16 DIABOLO"          PRG
33    "▪17 RITTER"           PRG
26    "▪18 NASHORN"          PRG
31    "▪19 PIONEER"          PRG
0     "----------------"     DEL
0     "------ENDE------"     DEL
0     "----------------"     DEL
0 BLOCKS FREE.
```

WICHTIGE HINWEISE

zur beiliegenden Diskette:

Aus den Erfahrungen der bisherigen Sonderhefte mit Diskette wollen wir ein paar Tips an Sie weitergeben:

1 **Bevor Sie mit den Programmen auf der Diskette arbeiten, sollten Sie unbedingt eine Sicherheitskopie der Diskette anlegen. Verwenden Sie dazu ein beliebiges Kopierprogramm, das eine komplette Diskettenseite dupliziert.**

2 **Auf der Originaldiskette ist wegen der umfangreichen Programme nur wenig Speicherplatz frei. Dies führt bei den Anwendungen, die Daten auf die Diskette speichern, zu Speicherplatz-Problemen. Kopieren Sie daher das Programm, mit dem Sie arbeiten wollen, mit einem File-Copy-Programm auf eine leere, formatierte Diskette und nutzen Sie diese als Arbeitsdiskette.**

3 **Die Rückseite der Originaldiskette ist schreibgeschützt. Wenn Sie auf dieser Seite speichern wollen, müssen Sie vorher mit einem Diskettenlocher eine Kerbe an der linken oberen Seite der Diskette anbringen, um den Schreibschutz zu entfernen. Probleme lassen sich von vornherein vermeiden, wenn Sie die Hinweise unter Punkt 2 beachten.**

ALLE PROGRAMME
aus diesem Heft







Diskette zum **Sonderheft** Nr. 63

Markt&Technik Verlag Aktiengesellschaft

Die auf diesem Datenträger enthaltenen Programme sind urheberrechtlich geschützt. Unerlaubte Kopierung, Vervielfältigung, Verleih oder Vermietung ist untersagt.

Anleitung siehe Rückseite ▶

IMPRESSUM

Herausgeber: *Carl-Franz von Quadt, Otmar Weber*

Redaktionsdirektor: *Dr. Manfred Gindle*
Chefredakteur: *Georg Klinge – verantwortlich für den redaktionellen Teil*
Stellv. Chefredakteur: *Arnd Wängler*
Textchef: *Jens Maasberg*
Produktion: *Georgia Sarikas*
Redaktion: *Harald Beiler (bl), Herbert Großer (gr)*
Mitarbeiter dieser Ausgabe: *Nikolaus Heusler*
Redaktionsassistenz: *Sylvia Derenthal, Diana Moser*
Telefax: *089/4613-5001*
Alle Artikel sind mit dem Kurzzeichen des Redakteurs und/oder mit dem Namen des Autors/Mitarbeiters gekennzeichnet

Manuskripteinsendungen: *Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt&Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt&Technik Verlag AG verlegten Publikationen. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.*

Verlagsleitung: *Wolfram Höfler*
Operation Manager: *Michael Koeppe*

Art-director: *Friedemann Porscha*
Layout: *Marian Schwarz*
Bildredaktion: *Sabine Tennstaedt; Roland Müller (Fotografie); Ewald Standke, Norbert Raab (Spritzgrafik); Werner Nienstedt (Computergrafik)*

Anzeigendirektion: *Jens Berendsen*
Anzeigenleitung: *Philipp Schiede (399) – verantwortlich für die Anzeigen*
Telefax: *089/4613-775*

Anzeigenverwaltung und Disposition: *Monika Burseg (147)*

Auslandsrepräsentation:
Auslandsniederlassungen:
Schweiz: *Markt&Technik Vertriebs AG, Kollerstr. 37, CH-6300 Zug, Tel. 042-440550/660, Telefax 042-415770*
USA: *M&T Publishing Inc.; 501 Galveston Drive Redwood City, CA 94063, Telefon: (415) 366-3600, Telex 752-351*
Österreich: *Markt&Technik Ges. mbH, Große Neugasse 28, A 1040-Wien Telefon: 0222/5871393, Telex: 047-132532*
Anzeigen-Auslandsvertretung:
England: *F. A. Smyth&Associates Limited, 23a, Aylmer Parade, London, N2 OPQ. Telefon: 0044/1/3405058, Telefax: 0044/1/3419602*

Israel: *Baruch Schaefer, Haeskel-Str. 12, 58348 Holon, Israel, Tel. 00972-3-5562256*
Taiwan: *Aim International Inc., 4F.-1, No. 200, Sec. 3, Hsin-I Rd.; Taipei, Taiwan, R.O.C., Tel. 00886-2-7548631, -7548633, Fax 00886-2-7548710*
Korea: *Young Media Inc., C.P.O. Box: 6113, Seoul/Korea, Tel. 0082-2-7564819, /-7742759, Fax 0082-7575789*
USA: *M&T Publishing Inc.; 501 Galveston Drive Redwood City, CA 94063, Telefon: (415) 366-3600, Telex 752-351*

Vertriebsdirektor: *Uwe W. Hagen*
Vertriebsmarketing: *Petra Schlichthärle (703)*

Vertrieb Handel: *Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: ip Internationale Presse, Ludwigstraße 26, 7000 Stuttgart 1, Tel. 0711/619660*

Bezugsmöglichkeiten: *Leser-Service: Telefon (089) 4613-366. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.*

Verkaufspreis: *Das Einzelheft kostet DM 16,–*

Produktion: *Technik: Klaus Buck (Ltg./180), Wolfgang Meyer (Stellv./887); Herstellung: Otto Albrecht (Ltg./917)*

Druck: *SOV Graphische Betriebe, Laubanger 23, 8600 Bamberg*

Sonderdruck-Dienst: *Alle in dieser Ausgabe erschienenen Beiträge sind in Form von Sonderdrucken zu erhalten. Anfragen an Reinhard Jarczok, Tel. 089/4613-185, Fax 4613-774.*

Vorstand: *Otmar Weber (Vors.), Bernd Balzer*

Direktor Zeitschriften: *Michael M. Pauly*

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen: *Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 522052, Telefax 089/4613-100*

ISSN 0931-8933

Telefon-Durchwahl im Verlag:
Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089/4613 und dann die Nummer, die in den Klammern hinter dem jeweiligen Namen angegeben ist.

Copyright-Erklärung



Name: ....

Anschrift: ....

Datum: ....

Computertyp: ....

Benötigte Erweiterung/Peripherie: ....

Datenträger: Kassette/Diskette ....

Programmart: ....

Ich habe das 18. Lebensjahr bereits vollendet

...., den ....

....
(Unterschrift)

Wir geben diese Erklärung für unser minderjähriges Kind als dessen gesetzliche Vertreter ab.

...., den ....

....

....

Bankverbindung:

Bank/Postgiroamt: ....

Bankleitzahl: ....

Konto-Nummer: ....

Inhaber des Kontos: ....

Das Programm/die Bauanleitung: ....

das/die ich der Redaktion der Zeitschrift 64'er übersandt habe, habe ich selbst erarbeitet und nicht, auch nicht teilweise, anderen Veröffentlichungen entnommen. Das Programm/die Bauanleitung ist daher frei von Rechten anderer und liegt zur Zeit keinem anderen Verlag zur Veröffentlichung vor. Ich bin damit einverstanden, daß die Markt & Technik Verlag AG das Programm/die Bauanleitung in ihren Zeitschriften oder ihren herausgegebenen Büchern abdruckt und das Programm/die Bauanleitung vervielfältigt, wie beispielsweise durch Herstellung von Disketten, auf denen das Programm gespeichert ist, oder daß sie Geräte und Bauelemente nach der Bauanleitung herstellen läßt und vertreibt bzw. durch Dritte vertreiben läßt.

Ich erhalte, wenn die Markt & Technik Verlag AG das Programm/die Bauanleitung druckt oder sonst verwertet, ein Pauschalhonorar.



Keine umständlichen Ladeanweisungen und ein übersichtliches Inhaltsverzeichnis der Diskette auf dem Bildschirm – wer will das nicht? Unser »Disklader« macht es möglich.

Einige Arbeitsschritte sind notwendig, um beim C64 ein Inhaltsverzeichnis von der Diskette zu erhalten. Ferner erschweren diverse Unterdateien zu einem Programm die Übersicht im »Directory«. Genau hierfür finden Sie eine Hilfe auf der ersten Seite der beiliegenden Diskette – den »Disklader«. Er generiert eine eigene Benutzeroberfläche für Ihren C64. In ihr sind Funktionen integriert wie die Anwahl einzelner Programme (mit jeweiliger Kurzbeschreibung), automatisches Laden und Starten von Diskette oder Erkennung der richtigen Diskette und Diskettenseite.

Da sich der Disklader an erster Stelle (auf der ersten Seite) der beiliegenden Diskette befindet, genügt es, wenn Sie zum Laden eingeben:

```
LOAD":*",8
```

Nach der Bestätigung mit <RETURN> dauert es ca. 15 s, bis die Datei im Speicher ist. Sie starten mit RUN und <RETURN>. Anschließend wird das File entpackt (ca. 2 s) und es erscheint die Benutzeroberfläche des »Disklader« (Bild 1). In der rechten unteren Bildschirmhälfte sehen Sie in einer weißen Umrandung den Namen des ausgewählten Programmes. Die unterste Bildschirmzeile zeigt die dazugehörige Kurzerklärung. Zusätzlich finden Sie in der rechten unteren Bildschirmhälfte den Text »Seite 1 auf Disk« oder »Seite 2 auf Disk«. Da Sie die Inhaltsverzeichnisse beider Seiten (ohne die Disk zu wenden) durchblättern können, finden Sie hier die Angabe, auf welcher Seite der Diskette sich das gewählte Programm befindet.

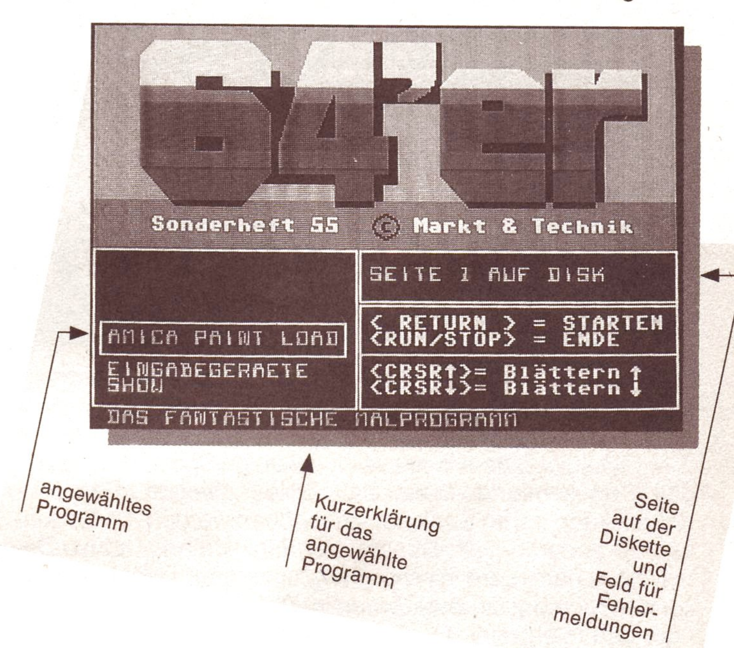
Durch Tastendruck <CRSR aufwärts> bzw. <CRSR abwärts> wählen Sie das nächste oder vorherige Programm von Diskette. Sie blättern sozusagen durch den Inhalt der Programme. <HOME> bringt Sie zum ersten Eintrag des Inhaltsverzeichnisses. Selbstverständlich sind nur die Programme verzeichnet, die sich eigenständig laden oder starten lassen.

<RETURN> führt Sie in den Ladeteil. Zuerst wird die »BAM« (Belegungsplan) der Diskette abgefragt und danach eine Fehlerkanalabfrage der Diskettenstation durchgeführt. Ist dieser Schritt in Ordnung, erscheint kurzzeitig »00, OK, 00,00« am Bildschirm. Eventuelle Fehleranzeigen bleiben sichtbar am Bildschirm (z.B. »21,READ ERROR,18,00« = Drive not ready). Sie lassen sich durch einen beliebigen Tastendruck wieder löschen. Schlagen Sie bitte vorher im Handbuch Ihrer Floppy nach und beseitigen Sie den Fehler. Eine andere Art der Fehlermeldung wird durch einen blinkenden Text dargestellt (z.B. »Bitte Disk wenden« oder »Falsche Diskette«). Sind keine Fehler aufgetreten, lädt der Disklader das von Ihnen gewählte Programm von der Diskette und startet es. Ladefehler, die in dieser Phase auftreten, werden nicht mehr berücksichtigt. Dieser Kompromiß mußte eingegangen werden, da manche Programme den Disklader überschreiben. Ein Rücksprung in den Lader (für die Fehlerabfrage) würde daher zum Absturz des Systems führen. Oder wir könnten nur Programme veröffentlichen, die mit der Benutzeroberfläche zusammenarbeiten. Viele Spiele, Tricks, oder Tools könnten nicht mehr geladen werden.

Für Sie bedeutet dies allerdings, nach jedem Starten eines Programmes den »Disklader« neu zu laden. Falls Sie die Benutzeroberfläche verlassen wollen, ist dies mit <RUN/STOP> möglich. Sie befinden sich dann im normalen »Basic« des C64. Für einen Neustart geben Sie

```
SYS 12032
```

ein und bestätigen mit <RETURN>. Dieser Neustart funktioniert auch nach einem »Reset«, d.h. wenn Sie durch einen angebrachten Taster einen »Hardware-Reset« ausgelöst haben. Sie sollten zwischenzeitlich kein Programm



geladen haben, da dies den verwendeten Speicherbereich überschreiben könnte. Laden Sie in diesem Falle den Disklader neu. Wir haben bei der Programmierung größten Wert auf Kompatibilität mit den unterschiedlichsten Betriebssystemerweiterungen gelegt. Lediglich bei der Gerätekonfiguration C128 mit RAM-Erweiterung und zweiter Diskettenstation sollten Sie die externe Floppy ausschalten. (gr)

Kurzinfo: Disklader

Programmart: Hilfsprogramm zum Laden der Programme auf der beiliegenden Diskette
Laden: LOAD":*",8
Starten: Nach dem Laden RUN eingeben
Steuerung: über Tastatur
Programmautor: H. Großer

Deluxe Datas - Assemblercode als Basic-Zeilen ausgeben

Datenexpreß

Gute Basic-Programme sind ohne Zusatzroutinen in Maschinensprache undenkbar. Schwierig ist nur, beide Programmierstile unter einen Hut zu bringen. »Deluxe Datas« schafft's im Handumdrehen.

von Jörg Lüttgau

Wer kennt es nicht, das Problem, Maschinensprache-Code mit einem Basic-Programm zu verbinden? Da gibt's zwei Möglichkeiten: Entweder lädt man das Maschinenprogramm gleich zu Beginn nach oder hängt es ans Basic-Programm in Form von Data-Zeilen, die innerhalb einer FOR-NEXT-Schleife an die betreffenden Speicherstellen gePOKEt werden. Vorteil der zweiten Methode: Die Diskette wird nur mit dem Platz für ein Programm (statt für zwei Dateien) belegt.

```
deluxe datas by j.luettgau
enter start-adress of data-maker
                  49152
program transfered to 49152-50367
using:
sys49152(,a$),a,b(,c,d,e,f)
a$=string ('data' or '.byte' e.g.)
a=start,b=end,c=datas/line
d=first line,e=step
f<>0 = checksum, g=value of $01
ready.
```

[1] Der Data-Generator ist startbereit: Auf dem Bildschirm erscheinen Hinweise zur Handhabung der Parameter.

Mit »Deluxe Datas« lassen sich beliebige Speicherbereiche des C64 komfortabel als Data-Zeilen ablegen. Das Programm kann frei im Speicher verschoben werden. Im Gegensatz zu anderen Data-Generatoren funktioniert »Deluxe Datas« nicht nach dem Prinzip »Programmierter Direktmodus«, sondern erzeugt die Data-Zeilen im Basic-RAM ohne Umweg über den Interpreter: Der Geschwindigkeitsvorteil ist enorm.

Durch die große Anzahl möglicher Parameterangaben ist »Deluxe Datas« sehr flexibel und für Assemblerprogrammierer ebenso interessant. Man kann z.B. das Befehlswort zu Beginn jeder Basic-Zeile neu definieren (statt »Data« etwa die Assembleranweisung ».BYTE«, mit der Tabellen zu einem Assembler-Quellcode eingeleitet werden usw.). Der Inhalt von Adresse 1 läßt sich ebenfalls ändern, wodurch das RAM unter dem ROM oder der Originalzeichensatz ausgelesen und in Data-Zeilen umgewandelt wird.

Sie laden das Utility mit

```
LOAD "DELUXE DataS",8
```

und starten es mit RUN. Zunächst müssen Sie eine bis zu fünfstellige Adresse im Dezimalformat angeben, an die Sie das 1215 Byte lange Utility verschieben wollen. Bei der Wahl der Adresse ist darauf zu achten, daß das Programm keine Speicherbereiche überschreibt, die Sie in Basic-Data-Zeilen umwandeln möchten, ebenso darf nicht der Bereich gewählt werden, den »Deluxe Datas« nach dem Laden und Starten selbst belegt: $0801 (2049) bis $0F52 (3922). Außerdem ist nicht erlaubt, das Maschinenprogramm ins RAM unters ROM zu transferieren. War die Verschiebeaktion erfolgreich, springt »Deluxe Datas« zurück in den Direktmodus. Sie erhalten die Meldung, an welcher Stelle im Speicher es sich jetzt befindet (Anfangs- und Endadresse, Abb. 1). Denken Sie auch daran, daß Data-Zeilen immer ab dem Basic-Anfang ($0801, dez. 2049) erzeugt werden. Je nach umzuwandelnder Datenmenge sollte »Deluxe Datas« möglichst hoch im Speicher angesiedelt werden. Unsere Empfehlung: Adresse 38000.

Der Aufruf der Programmfunktion mit den entsprechenden Parametern lautet:

```
SYS adresse(, string), start, ende(, dpz, zeile,
step, sum, val1)
```

Alle Parameter in Klammern können, müssen aber nicht angegeben werden. Die Bedeutung der Zahlen im einzelnen (alle Werte sind dezimal anzugeben):

adresse
SYS-Adresse zum Programmaufruf, an die »Deluxe Datas« verschoben wurde.

string
Wahlweise »Data« (für Basic-Zeilen) oder z.B. ».BYTE« (als Assembler-Quellcode). Besteht »string« aus Klartext, muß das Wort in Anführungszeichen stehen.

start, ende
Beginn und Ende des zu lesenden und umzuwandelnden Bereichs.

dpz
Anzahl der Data-Werte, die in einer umgewandelten Programmzeile stehen sollen (Idealwert: 16).

zeile, step
Bezeichnet die Nummer der ersten Data-Zeile, die weiteren folgen im Abstand »step«.

sum
Ist »sum« größer als »0«, wird eine Prüfsumme für die aktuelle Zeile gebildet (alle Werte addiert).

val1
Legt fest, ob aus dem RAM unterm ROM gelesen werden soll (Beispiel: der Wert »51« und die Adreßangaben ab 53248 bewirken das Lesen des Original-Commodore-Zeichensatzes).

»Deluxe Datas« eignet sich beispielsweise hervorragend zum Umwandeln von Sprite- und Hires-Speicherbereichen in Basic-Data-Zeilen. Vergessen Sie dabei jedoch nicht das Speicherplatzverhältnis »Maschinensprache« zu »Basic«: 8 KByte Assemblercode (z.B. eine Hires-Grafik) belegen nach der Umwandlung in Basic-Data-Zeilen weit über 100 Blocks auf Diskette. Die erzeugten Data-Zeilen lassen sich dann in Basic wie gewohnt innerhalb einer FOR-NEXT-Schleife mit READ und POKE wieder als Maschinencode ablegen. (bl)

Kurzinfo: Deluxe Datas

Programmart: Utility zum Umwandeln beliebiger Speicherbereiche in Basic-Datazeilen.
Laden: LOAD "DELUXE DATAS",8
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur
Besonderheiten: Programm im Speicher frei verschiebbar. Unsere Empfehlung: entweder weit oben im Basic-RAM, z.B. $9470 (38000) oder im Bereich von $C000 (49152) bis $CFFF (53247).
Benötigte Blocks: 8
Programmautor: Jörg Lüttgau



Farbige Tools

COLOR IS EASY

Es gibt die verschiedensten Grafikanwendungen für Ihren C64. Viele davon sind so speziell, daß man sie selbst programmieren muß. Für alle diejenigen bieten wir eine Fülle von Tools.

Grafikerweiterung für den Lores-Bildschirm

Ist es Ihnen nicht auch schon passiert? Zu einem Rechenergebnis wäre zur Erläuterung eine Kurvendarstellung mehr als sinnvoll gewesen. Sie haben auch schon von der Blockgrafik gehört. Doch das Zeichnen von Linien oder Kurven wird in Basic zu langsam. Also entfällt diese Darstellung. Das muß in Zukunft nicht mehr so sein. Unsere kleine Basic-Erweiterung erlaubt es, mit einem einfachen SYS-Aufruf im normalen Textmodus einen Pixel (4 x 4 Punkte pro Buchstabe) zu setzen. Zuerst muß allerdings das Programm mit

```
LOAD"LORES",8,1
```

geladen werden. Anschließend sollten Sie NEW und <RETURN> eingeben. Bei der Verwendung in einem eigenen Programm empfiehlt sich zu Beginn folgende Basic-Zeile:

```
10 IF A = 0 THEN A = 1: LOAD"LORES",8,1
```

Danach können Sie beliebig weiterprogrammieren. Das Nachladen des Tools ist gesichert. Vergessen Sie aber nicht, »Lores« mit auf die Programmdiskette zu kopieren.

Benützt wird das kleine Tool mit folgender Syntax:

```
SYS 49152,x-Wert,y-Wert,Modus,Farbe
```

»x-Wert« stellt die horizontale Achse dar und kann Werte von 0 bis 79 annehmen.

»y-Wert« ist die vertikale Achse mit möglichen Werten von 0 bis 49.

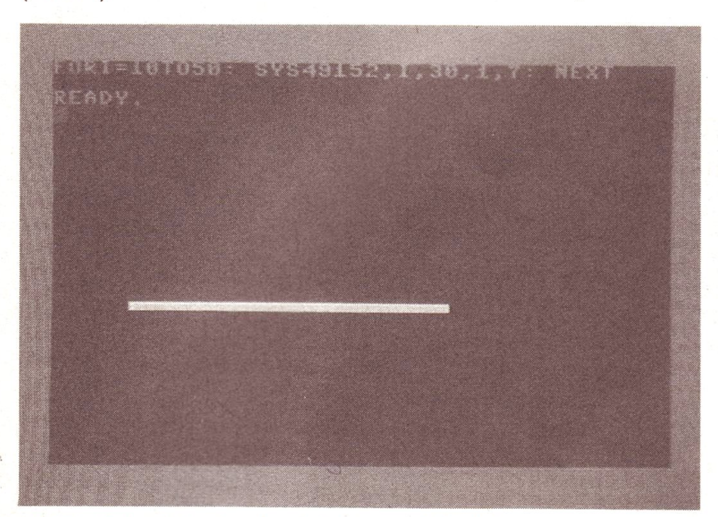
»Modus« gibt an, ob der Punkt gesetzt (1), gelöscht (0) oder getestet (255) werden soll. Beim Test steht das Ergebnis (1=gesetzt, 0=gelöscht) in Speicherzelle 2.

»Farbe« ist der normale Farb-POKE (0 bis 15). Wenn auf »Punkt gesetzt/gelöscht« geprüft werden soll, muß Farbe entfallen.

Dazu ein Beispiel:

```
FORI=10TO50:SYS 49152,I,30,1,7:NEXT
```

setzt eine gelbe Linie ab Spalte 10 bis Spalte 50 auf Zeile 30 (Abb. 1).



[1] Ein Beispiel, wie eine Linie gesetzt wird

Kurzinfo: Lores-Bildschirm

Programmart: Blockgrafik-Tool
Laden: LOAD"LORES",8,1
Starten: siehe Anleitung oben
Benötigte Blocks: 2
Programmautor: Georg Brandt

Laufschrift im Bildschirmrahmen

Seit einiger Zeit ist es in Mode gekommen, im Titelbild der Programme mindestens eine Laufschrift darzustellen. Nachdem sich die Programmiertechnik bis zur Flimmerfreiheit weiterentwickelt hat, bleibt immer noch ein Problem: Wohin, ohne den eh schon knappen Textbildschirm zu beschneiden?

Ganz einfach, an den ungenutzten Bildschirmrand. Er konnte lange Zeit immer nur die Farbe annehmen, die in $D020 angegeben war. Vor einiger Zeit kam jemand auf die Idee, diesen Platz dem VIC durch geschicktes Umschalten in den 24-Zeichen-Modus (durch Löschen des Bit ROW SELECT im Register $D011) als nutzbaren Raum zu entlocken. Die Euphorie über diese Entdeckung wurde jedoch rasch wieder gebremst: Außer dem einen Byte am Ende des VIC-Adreßraums, das nun langweilige Streifen erzeugt, kann der VIC in diesem Bereich nur Sprites darstellen. So lag die einzige Nutzung dieses zusätzlichen Raums darin, ein paar einsame Sprites über den Rand hinauswandern zu lassen. Diese können aber auch sinnvoller eingesetzt werden! Bei einigermaßen geschicktem Einsatz der acht zur Verfügung stehenden Sprites läßt sich eine brauchbare Laufschrift auf dem Rahmen erzeugen. Es sind aber durchaus auch zwei Laufschriften möglich. Genau dies tut unser Programm. Es scrollt acht Sprites, deren Inhalt in $3E00 bis $3FFF abgelegt sein muß, von rechts nach links über den Bildschirm. Das dazugehörige kleine Basic-Programm wird mit

```
LOAD"BASIC-DEMO",8
```

geladen und mit Run gestartet. Es berechnet aus den eingegebenen Texten die Daten für diese Sprites und schreibt sie gleich in den richtigen Bereich (Abb. 2).

Nachdem das Basic-Programm etwa zehn Sekunden lang gerechnet hat, erscheinen die eingegebenen Texte als Laufschriften ober- und unterhalb des Bildschirms. Erneutes Einschalten ist nach <RUN/STOP RESTORE> durch SYS 49152 möglich, da die Daten im Speicher erhalten bleiben.

Achtung: Vor der Startwiederholung des Basic-Teils muß die Interrupt-Routine durch <RUN/STOP RESTORE> oder durch Eingabe von

```
SYS 58648:SYS 64789:SYS 64931
```

(klappt nicht immer) abgeschaltet werden.

Mögliche Änderungen des Programms: Am einfachsten läßt sich die Farbe der Schrift ändern. Hierzu ruft man die Routine nur mit einer anderen Cursor-Farbe auf. Zum Verschieben der Laufschriften in y-Richtung muß der Inhalt der Speicherzelle 49174 verändert werden (normal = 8). In Adresse 49206 steht die Blocknummer des letzten Sprite (normal = 255). Wird diese verändert, müssen natürlich auch im Basic-Teil die Zahlen 15911 und 15872 in Zeile 270/280 verändert werden. Auch mit einem geänderten Zeichensatz läßt sich die Laufschrift einsetzen. Dazu muß die Variable ZA in Zeile 180 auf den Beginn des neuen Zeichensatzes zeigen. Setzt man hier 55296 ein, erscheint die Laufschrift in Kleinschrift. Das Hauptprogramm wird mit

```
LOAD"LAUFSCHRIFT",8,1
```

geladen. Experimentieren Sie ruhig mit den Werten in Zeile 310 des Demoprogramms.

[2] Das Basic-Demo zeigt die Laufschrift

Kurzinfo: Laufschrift

Programmart: Laufschriftgenerator
Laden: LOAD"BASIC-DEMO",8
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur
Besonderheiten: lädt »LAUFSCHRIFT« nach
Benötigte Blocks: 4
Programmautor: N. Nebel

Screen-Tool

Screen-Tool umfaßt einige Routinen, die das Arbeiten mit dem Bildschirm etwas erleichtern.

1. Cursor setzen:
   SYS 49500, Zeilen-Nr., Spalten-Nr.
2. Zeilen löschen
   SYS 49503, erste Zeile, letzte Zeile
3. Bildschirm invertieren/reinvertieren
   SYS 49506, Zeilen-Nr., Spalten-Nr., Anzahl Zeichen
   SYS 49521, Zeilen-Nr., Spalten-Nr., Anzahl Zeichen

Kurzinfo: Screen-Tool

Programmart: erweiterte Bildschirmroutinen
Laden: LOAD"SCREEN-TOOL",8
Steuerung: siehe Beschreibung
Benötigte Blocks: 1

Entgegen der normalen Betriebssystempraxis beginnt Screen-Tool bei der Numerierung der Zeilen bzw. Spalten nicht mit 0, sondern mit 1.

Geladen wird das Programm mit
```
LOAD"SCREEN-TOOL",8,1
```
Danach empfiehlt es sich, im Direktmodus NEW und <RETURN> einzugeben. Im Programm erübrigt sich dieses bei einer Nachladezeile (siehe Zeile 10 bei »Laufschrift«).

Spriteset

Spriteset setzt die Parameter für ein Sprite, schaltet es aber nicht ein. Geladen wird mit
```
LOAD"SPRITESET",8,1
```
Auch hier empfiehlt sich im Direktmodus ein NEW.

```
POKE53269,1:SYS49700,0,11,100,50,6,1
READY.
```

[3] Das Beispielsprite

Bevor Sie Spriteset verwenden, müssen Sie das (die) Sprite(s) einschalten:
```
POKE53269,1
```
Diese Anweisung aktiviert Sprite »0«. Es ist allerdings noch nicht sichtbar, da nicht alle Parameter stimmen. Um diese zu setzen, gilt folgendes Format:

SYS 49700, Sprite-Nr., Block-Nr., x-Pos, y-Pos, Farbe, Expand (jn)

Beispielsweise setzt
```
SYS 49700,0,11,100,50,6,1
```
für Sprite Nr. 0 den Block 11 (ab 704) als Datenspeicher fest, positioniert ihn auf die Koordinaten 100/50, färbt ihn blau und vergrößert ihn in x- und y-Richtung (Abb. 3).

Kurzinfo: Spriteset

Programmart: bewegt Sprites
Laden: LOAD"SPRITESET",8
Besonderheiten: Sprites müssen vorher eingeschaltet werden
Benötigte Blocks: 1

Werte grafisch dargestellt

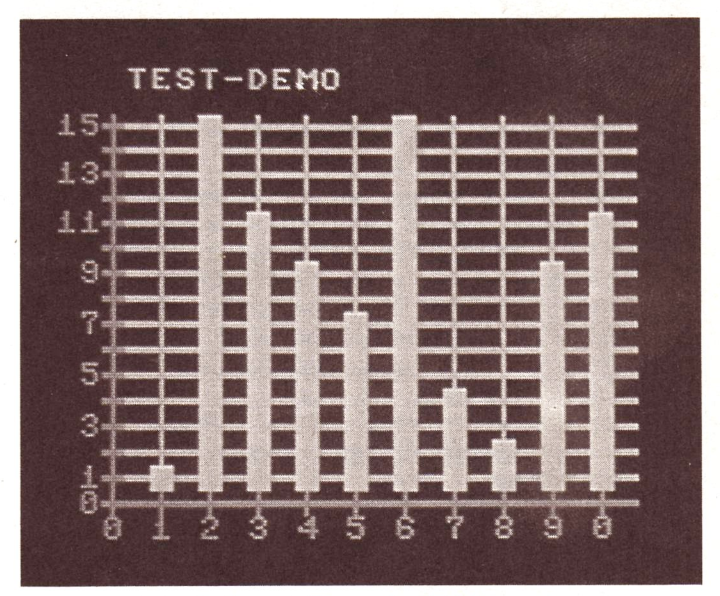
Damit Ihre Programme aussagekräftiger werden, finden Sie hier ein Unterprogramm, mit dem Werte als Balkendiagramm dargestellt werden können.

Grafische Darstellung und Bilder helfen, Werte klarer und schneller aufnehmen zu können, als dies nüchterne Zahlen können. Die Routine »Balkenstatistik« eignet sich zum Einbau in eigene Programme, bei denen es darauf ankommt, Werte leicht erfaßbar in grafischer Form darzustellen.

Nachdem Sie das Programm mit
```
LOAD"BALKENSTATISTIK",8
```
geladen und mit RUN gestartet haben, verlangt die Routine von Ihnen die Eingabe der Werte. Dabei ist zu beachten, daß bei Bildschirmausgabe die x-Einteilung nicht höher als 17 Einheiten sein darf. Die Anzahl der Balken beschränkt sich auf 16. Bei den Werten können beliebige Größen eingegeben werden, die aber nicht größer als die x-Einteilung sein sollten.

Nachdem das Diagramm berechnet und dargestellt wurde, kann man nun mit <N> eine neue Statistik erarbeiten, mit <D> die auf dem Bildschirm sichtbare Statistik (Abb. 4) auf einem MPS-Drucker ausgeben oder mit <E> das Programm beenden.

Wenn auf die Bildschirmausgabe verzichtet wird und die Werte gleich an den Drucker wandern sollen, erhöht sich die x-Einteilung auf 100, und die Anzahl der Balken kann nun 35 betragen. Auch der Text, der zusätzlich ausgegeben werden kann, darf nun bis zu 70 Zeichen lang sein.



[4] Anhand der Balkengrafik werden Zahlenwerte überschaubar

Kurzinfo: Balkenstatistik

Programmart: grafische Zahlenauswertung
Laden: LOAD"BALKENSTATISTIK",8
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur
Benötigte Blocks: 6
Programmautor: Daniel Wäschle

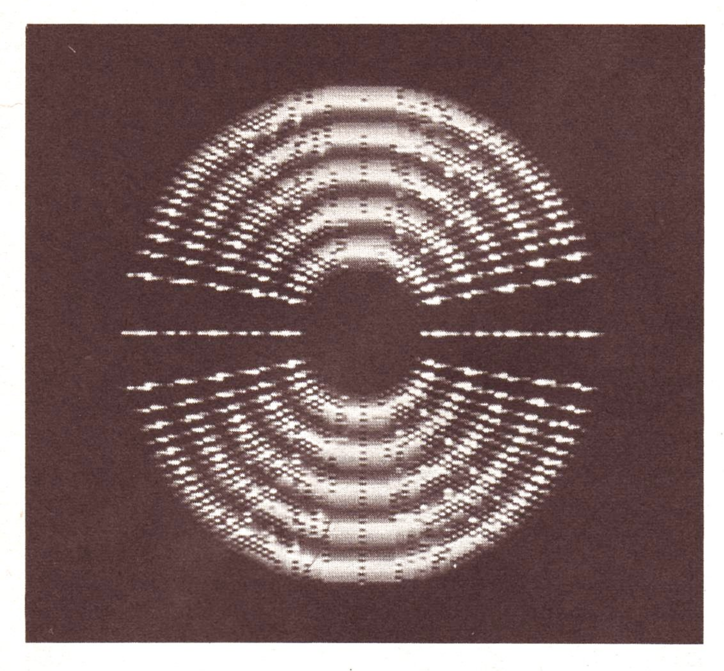
E.C.R.-Plot

Das Programm E.C.R.-Plot stellt dem C64 eine neue Farbauflösung zur Verfügung. Es erhöht die Farbfähigkeit des VIC folgendermaßen:

Bisher standen in einem 8 x 8 (bzw. Multicolor 4 x 8) großen Pixelfeld nur drei Farben zur Verfügung. Zwei aus einem extra definierten 1 KByte großen Farbspeicher und eine aus dem Color-RAM.

Der ECR-Mode schaltet nun jede Rasterzeile des Registers, in dem die Position des Farbspeichers angegeben ist, in 1-KByte-Schritten höher, um nach acht Schritten wieder von vorne zu beginnen.

Dadurch verkleinert sich der Machtbereich des Farbspeichers auf ein 8 x 1 (bzw. Multicolor 4 x 1) großes Pixelfeld. Resultat: In einem 8 x 8-Punktefeld können nun statt drei Farben (bzw. vier, wegen Hintergrundfarbe), 2 x 8 (Farbspeicher) + 1 Hintergrundfarbe + 1 Farbe aus Color-RAM = 18 Farben verwendet werden (der C64 besitzt nur 16!). Leider aber



[5] Mit ECR-Plot wird eine Farbenpracht erreicht, wie sie der C64 bisher nicht kannte

reicht das immer noch nicht, um wie z.B. beim Amiga jeden Punkt einzeln einzufärben, denn nebeneinander passen nur maximal drei unterschiedliche Farben auf eine Breite von vier Pixels. (Eigentlich können wir darüber froh sein, denn 8 KByte x 4 Pixels gibt 32 KByte. Das hieße, daß bei einem Bild die Hälfte des C-64-Speichers belegt wäre.) Aber auch der E.C.R.-Mode hängt hier nicht so hinterher, denn mit 17 KByte frißt der auch ganz schön was weg.

Warum 17 KByte?
8 KByte Farbspeicher (bisher 1 KByte)
8 KByte Bitmap
1 KByte Color-RAM

Zudem ist der E.C.R.-Mode etwas konfus, da die Farben nicht gerade der Reihe nach kommen:

1. Zeile - Farbspeicher 1 /Color-RAM 1 Zeile
2. Zeile - Farbspeicher 2 /Color-RAM 1 Zeile
3. Zeile - Farbspeicher 3 /Color-RAM 1 Zeile
4. Zeile - Farbspeicher 4 /Color-RAM 1 Zeile
5. Zeile - Farbspeicher 5 /Color-RAM 1 Zeile
6. Zeile - Farbspeicher 6 /Color-RAM 1 Zeile
7. Zeile - Farbspeicher 7 /Color-RAM 1 Zeile
8. Zeile - Farbspeicher 8 /Color-RAM 1 Zeile
9. Zeile - Farbspeicher 1, 2. Zeile /Color-RAM 2. Zeile

Eine ganz schöne Arbeit für eine Punktsetzroutine, erst mal die ganzen Adressen zu berechnen und als Krönung des ganzen dann noch einen Algorithmus abzuarbeiten, der bei Farbenüberschneidungen möglichst die unwichtigste Farbe eliminiert. So eine Plot-Routine ist bei E.C.R.-Pilot schon enthalten, ist auch ziemlich schnell unter normalen Umständen - und das ist genau der Punkt! Beim E.C.R.-Mode herrschen keine normalen Umstände mehr, denn durch den riesigen Raster IRQ gehen ca. 70 bis 80 Prozent der Prozessorzeiten flöten.

Geladen wird das Programm mit

```
LOAD"ECR - PLOT",8,1
```

Empfehlenswert ist der Programmstart allerdings nur in einem Programm, da keine Bildschirmausgaben mehr sichtbar sind.

Kurzinfo: E.C.R. Plot

Programmart: erweiterte Farbfähigkeiten
Laden: LOAD"ECR - PLOT",8
Steuerung: siehe Beschreibung
Benötigte Blocks: 6
Programmautor: Marcus Hillebrand

Bedienung von E.C.R.-Plot

Die Punktsetzroutine hat folgende Syntax:
sys 49152,x-Koordinate,y-Koordinate,Farbe (zum Löschen Farbe 0)
sys 49155 - aktiviert das ECR-Bild von $4000-$8000
sys 49158 - schaltet ECR-Bild ab
jsr $C009 - die Plot-Routine für Assemblerfreaks
x-Register -x-Koordinate
y-Register -y-Koordinate
Zeropage $21 - Farbe

Auf der beiliegenden Diskette befindet sich ein Demoprogramm (Abb. 5). Geladen wird es mit

```
LOAD"ECR-DEMO",8
```

Bevor Sie starten (mit RUN), sollten Sie den Bereich $4000 bis $8000 löschen. Am schnellsten geschieht dies in einem Monitor. Zum Beispiel »F 4000 8000 00« oder »O 4000 8000 00«, je nach Monitor.

Die Register des VIC

Im Handbuch des C64 finden Sie eine Beschreibung der VIC-Register. Sie ist leider nicht ganz vollständig, und es fehlt die Nennung der Normalwerte. Mit der Tabelle 1 steht Ihnen eine vollständige Übersicht aller Register des VIC zur Verfügung. (gr)

Die Register des VIC
Basisadresse 53248 ($D000)

Register	Normwert	Zweck
0 53248 $D000	0, wenn Sprite nicht verwendet	x-Koordinate von Sprite #0 ist der Wert < 24, dann ist der Sprite verdeckt.
1 53249 $D001	0, wenn Sprite nicht verwendet	y-Koordinate von Sprite #0 ist der Wert < 50 > 229, dann ist der Sprite verdeckt
2 53250 $D002	siehe 0	x-Koordinate von Sprite #1 sonst siehe 0
3 53251 $D003	siehe 0	y-Koordinate von Sprite #1 sonst siehe 0
4 53252 $D004	siehe 0	x-Koordinate von Sprite #2 sonst siehe 0
5 53253 $D005	siehe 0	y-Koordinate von Sprite #2 sonst siehe 0
6 53254 $D006	siehe 0	x-Koordinate von Sprite #3 sonst siehe 0
7 53255 $D007	siehe 0	y-Koordinate von Sprite #3 sonst siehe 0
8 53256 $D008	siehe 0	x-Koordinate von Sprite #4 sonst siehe 0
9 53257 $D009	siehe 0	y-Koordinate von Sprite #4 sonst siehe 0
10 53258 $D00A	siehe 0	x-Koordinate von Sprite #5 sonst siehe 0
11 53259 $D00B	siehe 0	y-Koordinate von Sprite #5 sonst siehe 0
12 53260 $D00C	siehe 0	x-Koordinate von Sprite #6 sonst siehe 0
13 53261 $D00D	siehe 0	y-Koordinate von Sprite #6 sonst siehe 0
14 53262 $D00E	siehe 0	x-Koordinate von Sprite #7 sonst siehe 0
15 53263 $D00F	siehe 0	y-Koordinate von Sprite #7 sonst siehe 0

Alle Register des VIC mit den Normalwerten

Die Register des VIC
Basisadresse 53248 ($D000)

Register	Normwert	Zweck
16 53264 $D010	siehe 0	9. Bit der x-Koordinaten für Pos. 256-319 zuständig: Bit: 7 6 5 4 3 2 1 0 Sprite: 7 6 5 4 3 2 1 0
17 53265 $D011	27	Kontrollregister für Modus des VIC Belegung: 0 1 weiches scrollen (0-7) 2 3 25/24 Zeilen 1/0 4 VIC ein/aus 1/0 5 HiRes/Text 1/0 6 Hintergrundfarb/Text 1/0 7 9.Bit Rasterwert
18 53266 $D012	wechselt	Nummer der gerade aktuellen Rasterzeile
19 53267 $D013	wechselt	x-Koordinate des Lightpen
20 53268 $D014	wechselt	y-Koordinate des Lightpen
21 53269 $D015	0, wenn Sprite nicht verwendet	Schaltet die Sprites aus/ein aus (0) / ein (0) Bit: 7 6 5 4 3 2 1 0 Sprite: 7 6 5 4 3 2 1 0
22 53270 $D016	200	Kontrollregister für erweiterte Funktionen 0 1 weiches scrollen (0-7) 2 3 40/38 Textspalten 1/0 4 Multicolor/Normal 1/0 5 immer 0 6 immer 1 7 immer 1
23 53271 $D017	0, wenn Sprite nicht verwendet	x-Vergrößerung Sprite # 0 klein (0) / groß (1) Bit: 7 6 5 4 3 2 1 0 Sprite: 7 6 5 4 3 2 1 0
24 53272 $D018	21	Speicheraufteilung des VIC 0 immer 1 1 2 Character-RAM/ROM 3 4 5 Bildschirm-RAM 6 7
25 53273 $D019	wechselt	IRQ Statusregister Ergebniss eines IRQ-Ereignis 0 Rasterzeile 1 Sprite-Hintergrundkollision 2 Sprite-Spritekollision 3 Lightpen 4 immer 1 5 immer 1 6 immer 1 7 Interrupt stattgefunden(1)
26 53274 $D01A	wechselt	IRQ Maske Die in Reg. 25 abzufragenden Ereignisse werden mit Bit=1 zugelassen 0 Raster-IRQ 1 Sprite-Hintergrund 2 Sprite-Sprite 3 Lightpen 4 immer 1 5 immer 1 6 immer 1 7 immer 1

Die Register des VIC
Basisadresse 53248 ($D000)

Register	Normwert	Zweck
27 53275 $D01B	0, wenn Sprite nicht verwendet	Priorität des Sprite 0 = vor, 1 = hinter dem Text Bit: 7 6 5 4 3 2 1 0 Sprite: 7 6 5 4 3 2 1 0
28 53276 $D01C	0, wenn Sprite nicht verwendet	Multicolor für Sprites 0 Schaltet aus, 1 schaltet ein Bit: 7 6 5 4 3 2 1 0 Sprite: 7 6 5 4 3 2 1 0
29 53277 $D01D	0, wenn Sprite nicht verwendet	y-Vergrößerung Sprite # 0 klein (0) / groß (1) Bit: 7 6 5 4 3 2 1 0 Sprite: 7 6 5 4 3 2 1 0
30 53278 $D01E	0, wenn Sprite nicht verwendet	Kollisionsregister Sprite-Sprite Bit= 0 wenn keine Kollision Bit: 7 6 5 4 3 2 1 0 Sprite: 7 6 5 4 3 2 1 0
31 53279 $D01F	0, wenn Sprite nicht verwendet	Kollisionsregister Sprite-Hintergrund Bit= 0 wenn keine Kollision Bit: 7 6 5 4 3 2 1 0 Sprite: 7 6 5 4 3 2 1 0
32 53280 $D020	254	Rahmenfarbe 0 = schwarz 8 = orange 1 = weiß 9 = braun 2 = rot 10 = hellrot 3 = cyan 11 = grau III 4 = rosa 12 = grau II 5 = grün 13 = hellgrün 6 = blau 14 = hellblau 7 = gelb 15 = grau I
33 53281 $D021	246	Bildschirmfarbe (Farben s. Reg. 32)
34 53282 $D022	241	Farbe 1 für Multicolor und erweiterter Hintergrundfarb-Modus (nur Text) (Farben s. Reg. 32)
35 53283 $D023	242	Farbe 2 für Multicolor und erweiterter Hintergrundfarb-Modus (nur Text) (Farben s. Reg. 32)
36 53284 $D024	243	Farbe 3 für Multicolor und erweiterter Hintergrundfarb-Modus (nur Text) (Farben s. Reg. 32)
37 53285 $D025	244	Farbe 1 für Multicolor-Sprite (Farben s. Reg. 32)
38 53286 $D026	244	Farbe 2 für Multicolor-Sprite (Farben s. Reg. 32)
39 53287 $D027	241	Farbregister Sprite #0 (Farben s. Reg. 32)
40 $D028 53288	242	Farbregister Sprite #1 (Farben s. Reg. 32)
41 $D029 53289	243	Farbregister Sprite #2 (Farben s. Reg. 32)
42 $D02A 53290	244	Farbregister Sprite #3 (Farben s. Reg. 32)
43 $D02B 53291	245	Farbregister Sprite #4 (Farben s. Reg. 32)
44 $D02C 53292	246	Farbregister Sprite #5 (Farben s. Reg. 32)
45 $D02E 53293	247	Farbregister Sprite #6 (Farben s. Reg. 32)
46 $D02F 53294	252	Farbregister Sprite #7 (Farben s. Reg. 32)

B.M.G. – Bildschirmmasken-Generator für Verwöhnte

Maskenball

Ab sofort wird das Entwerfen von Bildschirmmasken zum Kinderspiel: Mit dem komfortablen Editor »B.M.G« zeichnen Sie Masken beliebiger Art und Farbe, die in Basic-Zeilen umgewandelt werden.

von Roman Bartke

[1] Startbildschirm des »B.M.G«: Die RETURN-Taste aktiviert den Editiermodus, um eine Maske aufzubauen.

B.M.G.«, der Bildschirmmasken-Generator, geht neue Wege. Es kommt nicht so sehr auf die Arbeitsgeschwindigkeit an, sondern auf den Komfort nach der Masken-Umwandlung in Basic-Zeilen: FOR-NEXT- und SPC(x)-Befehle werden nur dann angewandt, wenn sie auch wirklich eine Speicherplatzersparnis bringen. Falls das letzte Zeichen (Speicherzelle 2023) vorhanden ist, wird auch dieses mit seiner Farbe in der letzten Zeile des generierten Programms gePOKEt. Leerzeichen bindet das Programm mit <CRSR abwärts> in die nächste Zeile ein, eine eventuelle Zeichensatzumschaltung wird ebenfalls berücksichtigt. Der neue Editor eignet sich hervorragend, um Blockgrafikbilder zu zeichnen. Es lassen sich Linien ziehen, Zeilen löschen und zentrieren, Spalten eliminieren, Hintergrund- und Rahmenfarben umschalten: Er ist bedeutend komfortabler als der Standardeditor des Basic 2.0!

Das Programm wird mit: `LOAD "B.M.G",8`
geladen und mit RUN gestartet. Zu Beginn erscheint die Frage: »Sollen "SHIFT/SPACE" den "SPACE"-Zeichen gleich sein?« (Abb. 1)

Falls Sie die SHIFT/SPACE-Taste nicht für andere Zwecke benötigen, dürfen Sie die RETURN-Taste drücken, ansonsten <N> und <RETURN>.

Das Titelbild läßt sich mit <SHIFT CLR/HOME< löschen, der Editor-Cursor befindet sich in der linken oberen Ecke des Bildschirms. Folgende Bedienungsmöglichkeiten stehen für den Entwurf einer Maske zur Verfügung:

- **alle in Basic 2.0 erreichbaren Tasten** außer CHR$(34)=»"« lassen sich zum Zeichnen verwenden,
- <F1>: Rahmenfarbe einstellen,
- <F3>: Hintergrundfarbe weiterschalten,
- <F5>: Wiederholfunktion für alle Tasten aktivieren,
- <F7>: Tastenwiederholung ausschalten,
- **alle Cursor-Tasten** in gewohnter Manier (rauf, runter, rechts, links),
- <CBM SHIFT>: Zeichensatz umschalten (Großschrift-Blockgrafik/Kelin-Großschrift),
- **Cursor-Farben:** wie in Basic 2.0, erreichbar durch die Tasten <CTRL> bzw. <CBM> plus entsprechende Zahlentaste,
- <CTRL Z>: Zeile zentrieren,
- <CTRL X>: Zeile löschen,
- <CTRL Y>: Spalte löschen,
- <CTRL L>: Linie zeichnen.

Dazu fährt man über das Zeichen, aus dem eine Linie bestehen soll, und betätigt <CTRL L>. Nun kann man sich mit Hilfe der Cursor-Tasten ans gewünschte Ende der Linie bewegen und <RETURN> drücken: Die Zeichenfolge wird ausgegeben, der Cursor befindet sich am Linienende.

- <RVS ON> und <RVS OFF>: wie in Basic 2.0.
- <INST>: fügt an der aktuellen Cursor-Position ein Leerzeichen (SPACE) ein und verschiebt die restliche Zeile nach rechts,
- <DEL>: Der Cursor bewegt sich ein Zeichen nach links. Das Zeichen, auf dem sich der Cursor vorher befand, erlischt durch die nunmehr eingerückte Zeile.
- <CTRL (Pfeil nach links)>: Die Maske wird in Basic-Zeilen umgewandelt.

Um verfolgen zu können, wie weit der »B.M.G« mit seiner Arbeit ist, wird das ausgelesene Zeichen auf dem Bildschirm durch ein weißes Sternchen überschrieben. Keine Angst, die von Ihnen kreierte Maske geht dabei nicht verloren.

Anschließend fragt das Programm nach Startzeile und Schrittweite der zu generierenden Basic-Zeilen. Sind die Eingaben akzeptabel (keine unzulässigen Werte), wird die Maske gelistet. Jetzt kann man sie ausprobieren, speichern und die erzeugten Basic-Zeilen nach Wunsch an eigene Programme hängen.

Programmbeispiele

Unter den Dateinamen »64'ER.MSK« und »ADRESS.MSK« finden Sie zwei Basic-Programme auf der Diskette, die mit »B.M.G« auf dem Bildschirm entworfen und in Basic-Zeilen umgewandelt wurden. Sie lassen sich wie jedes andere Basic-Programm laden und bearbeiten, wobei »ADRESS.MSK« zeigt, daß man nicht nur Grafikmasken entwickeln kann, sondern auch solche, die sich speziell für Anwendungen (z.B. eine Adreßdatei), eignen. (bl)

Kurzinfo: B.M.G.

Programmart: Maskengenerator
Laden: LOAD "B.M.G",8
Starten: nach dem Laden mit RUN
Steuerung: Tastatur
Besonderheiten: Berücksichtigt alle 16 Farben, wandelt Inhalt eines Textbildschirms in Basic-Zeilen um.
Benötigte Blocks: 59
Programmautor: Roman Bartke

Graphen einfach plotten mit »Funktionenhilfe«

Rasante Kurven

Einfach zu bedienen und trotzdem professionell und vielseitig – so könnte man diesen Funktionsplotter umschreiben. Neben der Darstellung beliebiger Funktionsgraphen werden Nullstellen, Extremwerte und Definitionslücken berechnet.

von Michael Suhr

Die »Funktionenhilfe« spricht u.a. Schüler und Studenten an, die sich bei der Kurvendiskussion das Leben etwas einfacher machen möchten. Das Programm ist menügesteuert, besticht durch seine einfache Benutzerführung und leistet dennoch mehr als vergleichbare Funktionsplotter. Da die Funktionsvorschrift nach den Rechenregeln eingegeben wird, entfällt die oft umständliche und fehlerträchtige Umwandlung in ein für den Basic-Interpreter verständliches Format. Definitionslücken der Funktion werden automatisch erkannt und auf Wunsch grafisch dargestellt. Nullstellen und Minima bzw. Maxima lassen sich zusätzlich berechnen. Alle Ergebnisse werden in mathematisch korrekter Notation auf dem Bildschirm ausgegeben, die Druckerausgabe ist mit Epson-kompatiblen Druckern (z.B. LX-800) und – nach einer kleinen Anpassung – auch mit MPS-Druckern möglich. Sollten Sie wider Erwarten einmal nicht mehr weiterwissen, drücken Sie einfach die Hilfetaste und erhalten stante pede eine ausführliche Erklärung der aktuellen Programmfunktion.

Das Programm läuft auf dem C64 mit Diskettenlaufwerk und dem C128 im 64'er-Modus. Laden Sie es von der beiliegenden Diskette mit

```
LOAD "START",8
```

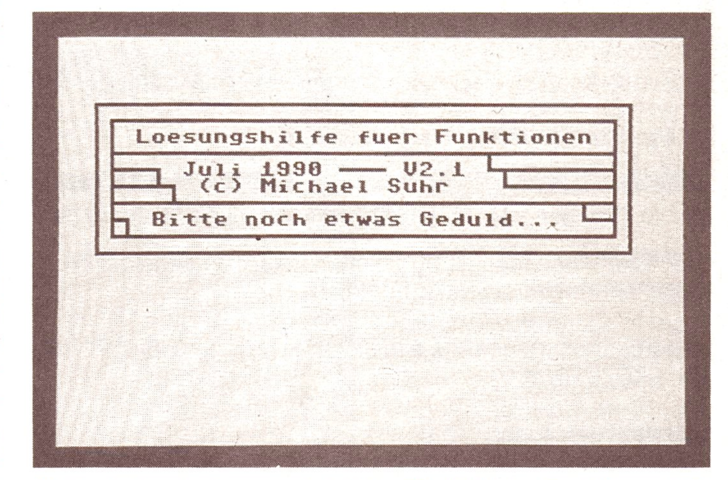
und geben Sie RUN ein. Während des Nachladens einiger Unterprogramme ist ein Titelbild zu sehen (Abb. 1). Nach wenigen Sekunden fordert Sie das Programm zur Eingabe der Funktion auf (Abb. 2). Verwenden Sie zur Eingabe die normale Basic-Syntax, mit folgenden Ausnahmen:

Funktionen

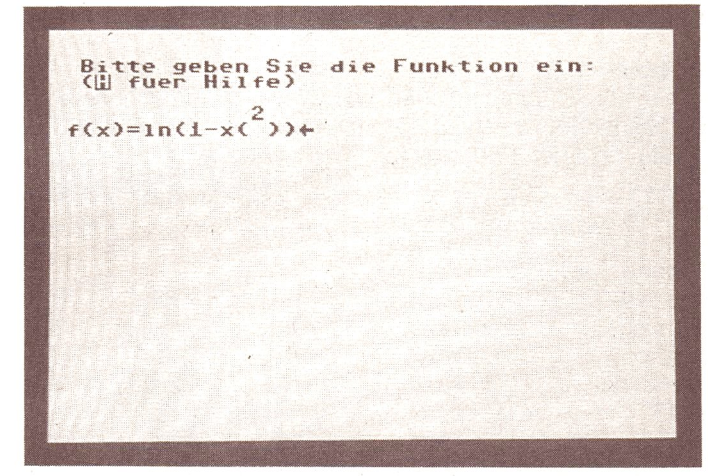
Hinter Funktionen können die Klammern entfallen, also SIN x statt SIN (x) oder AB X x statt ABS (x). Achten Sie aber auf Eindeutigkeit! So wird etwa der Term SIN 2x als (SIN 2) * x interpretiert. Für den Absolutwert (Betrag) können die Betragsstriche gesetzt werden, die Sie mit <SHIFT B> eingeben. Das Wurzelzeichen <SHIFT @> ist ebenfalls erlaubt, es ersetzt den Term für die Quadratwurzel SQR. Die Exponentialfunktion EXP (x) ist nicht erlaubt, schreiben Sie statt dessen e hoch x.

Sonderzeichen

Um die Eingabe möglichst komfortabel zu gestalten, werden auch verschiedene mathematische Sonderzeichen akzeptiert, z.B. wie oben erwähnt <SHIFT @> für das Wurzel-Symbol oder <SHIFT B> für den Betragsstrich. Für die Eulersche Zahl 2,718 281 828 steht e, und pi ergibt die Ludolphsche Zahl 3,141 592 654. Die Kleinbuchstaben außer x werden als Funktionsname betrachtet, x ist die Funktionsvariable. Neben den runden Klammern dürfen Sie auch eckige verwenden, beide werden vom Programm gleichgestellt.



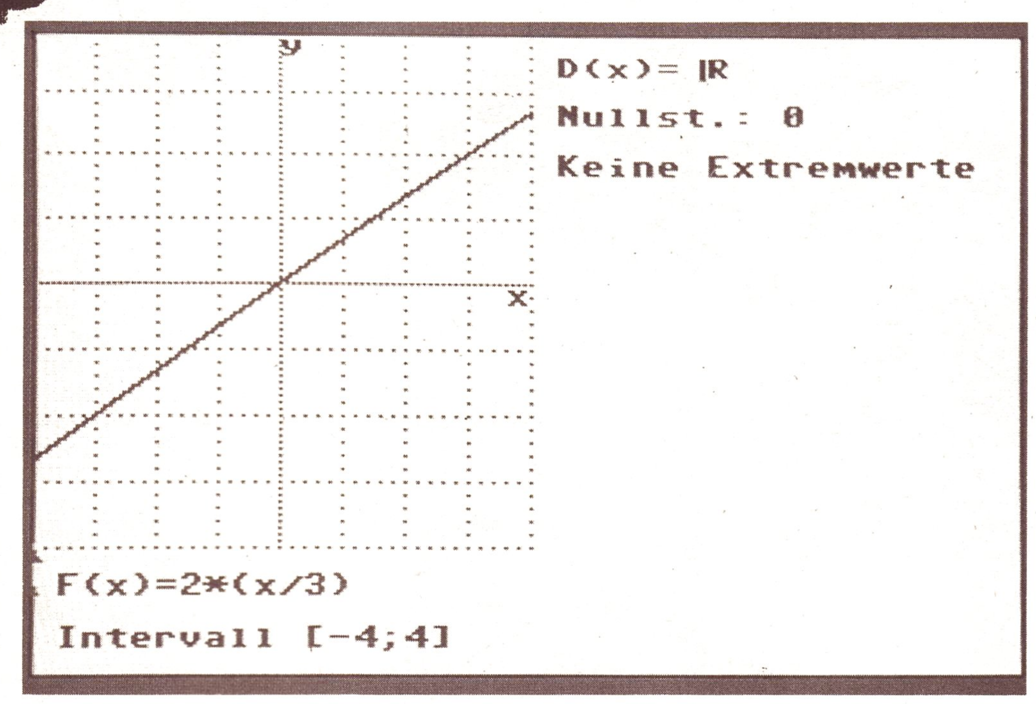
[1] Nach dem Laden erscheint das Titelbild der Funktionenhilfe



[2] Die Funktion kann so eingegeben werden, wie sie im Mathematikbuch steht

Potenzen

Potenzen können direkt eingegeben werden. Der Potenzmodus wird dabei mit der Taste <↑> eingeschaltet, mit <RETURN> kehren Sie wieder in den Normalmodus zu-



[4] Die Funktion wird im gewählten Intervall grafisch auf dem dem Bildschirm dargestellt

rück. Die Verschachtelung solcher Potenzen ist leider nicht vorgesehen. Um etwa die Funktion

```
Y = sin 2x hoch (1-x) - 15
```

zu berechnen, geben Sie ein: sin (2*x) (↑) 1-x <RETURN> -15 <RETURN>

Gleichungen lösen

Auch Gleichungen lassen sich mit dem Programm lösen. Anstelle des Gleichheitszeichens verwenden Sie das Minuszeichen; dort, wo die resultierende Funktion den Wert Null annimmt, ist die Gleichung gelöst (Nullstellensuche).

Beispiele

Um deutlicher zu machen, worauf bei der Funktionseingabe besonders zu achten ist, geben wir Ihnen hier einige Beispiele richtiger und falscher, weil mißverständlicher, Ausdrücke:

Falsch	Richtig
x9	9x oder 9*x
WurzelSINx	Wurzel(SINx)
x+3/x+4	(x+3)/(x+4) oder x+(3/x)+4
1/cosx	1/(cosx)
log x	lnx/(ln10)
(2-x)lnx	(2-x)*lnx
4+15	4+15*x
3.Wurzel x	x↑)(1/3)

Verwenden Sie bei der Eingabe keine Großbuchstaben! Zu vermeiden sind Funktionen, in denen die Variable x nicht vorkommt, die immer Null ergeben (z.B. x-x) oder Potenzfunktionen, die x selber ergeben (etwa x hoch 2). Wenn Sie die Funktionsvorschrift vollständig eingegeben haben, betätigen Sie <RETURN>. Das Programm bittet Sie daraufhin um Bestätigung und beginnt dann ggf. mit der Umwandlung der Funktion in ein für den Basic-Interpreter verständliches Format. Zugleich werden verschiedene Syntaxprüfungen vorgenommen. Dabei zählt und vergleicht das Programm die öffnenden und schließenden Klammern, wandelt e in EXP() und konvertiert den gesamten Term in Tokens. Falls Ihre Funktion einen Fehler enthalten sollte, meldet das Programm diesen mit einer kurzen Beschreibung. Sollte diese nicht ausreichen, drücken Sie die Taste <H>, die im gesamten Funktionenhilfeprogramm als Hilfetaste dient. Es wird dann ein kurzer erklärender Text von Diskette nachgeladen und angezeigt (Abb. 3 und Abb. 5). Mit <RETURN> werden ggf. die Textseiten weitergeblättert, mit <←> kehren Sie ins Hauptprogramm zurück.

Der Funktionsplotter

Wurde kein Fehler gefunden, wählen Sie als nächstes ein geeignetes Intervall. Entscheiden Sie sich für eine der vier Vorgaben: (-4,4), (-8,8), (-16,16) oder (-32,32). Das Intervall kann später im Programm geändert werden. Danach erscheint die Frage nach dem Zeichnen der Funktion. Wenn Sie sich hier für »ja« entscheiden und die <J>-Taste drücken, wird die Funktion geplottet, die Funktionsvorschrift sowie das gewählte Intervall in die Grafik geschrieben und alle

```
 f(x)=ln(1-x(^2))

 Funktion berechnen?

 Nein  (Zurueck)  --F1
 Ja    (Weiter )  --F7

Syntax II

  'f'

Ich fand ein unuebersetzbares Zeichen!

Diese Fehlermeldung erscheint immer
dann, wenn ein Zeichen keiner An-
weisung zugeordnet werden kann.
z.B.:

ln x; tan(x-k); |x-sqcx|

 Ende der Hilfestellung!

  Ein Tastendruck bitte...
```

[3] Auf Wunsch gibt das Programm umfangreiche Hilfstexte aus

später errechneten Werte ausgegeben. Sie haben zwei Möglichkeiten, das Zeichnen abzubrechen: Zum einen mit <←> - das Unterprogramm wird sofort verlassen. Zum anderen mit der <RETURN>-Taste, dann wird verfahren, als wäre das Intervall bis zum Schluß durchlaufen worden. Dies ist im folgenden für die Berechnungen der Nullstellen und der Extremwerte wichtig.

Beim Zeichnen schraffiert das Programm automatisch die Bereiche, in denen die Funktion nicht definiert ist. Wenn Sie dieses Feature ausprobieren wollen, lassen Sie sich die Funktion F(x)=ln(1-x(↑)2) im Bereich von -4 bis 4 darstellen. Der Term ist nur im Bereich von -1 bis 1 definiert, da das Argument der ln-Funktion positiv sein muß. Die verbleibenden Be-

reiche -4 bis -1 und 1 bis 4 werden grau unterlegt (schraffiert, Abb. 8).

Nach dem Zeichnen der Funktion fragt das Programm, ob das gewählte Intervall Ihnen noch zusagt. Bei Verneinung

Eine Funktion mit Lücken

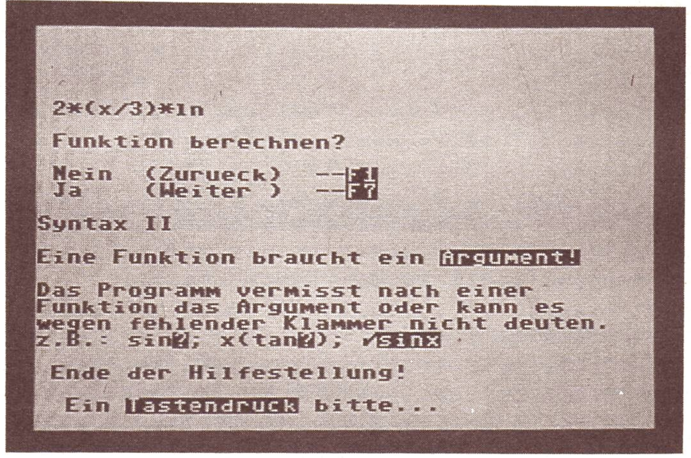
springt der Funktionsplotter zurück zur Intervallauswahl. Im anderen Fall erscheint die Frage, ob die Definitionslücken ermittelt werden sollen. Antworten Sie hier mit »Nein«, gelangen Sie sofort zum Hauptmenü. Das gleiche passiert, falls in der Funktion Beträge vorkommen. Aus programmtechnischen Gründen ist es nicht möglich, bei Betragsfunktionen den Definitionswert zu bestimmen. Bei »Ja« werden die Definitionslücken berechnet und in die Grafik eingetragen. Die Menge der rationalen Zahlen wird dabei mathematisch korrekt mit einem stilisierten R dargestellt. Durch die spätere Berücksichtigung der Ergebnisse dieser Berechnungen wird auch die Bearbeitung sonst kritischer Funktionen wie 1/x oder SQR(x) möglich.

Liegen keine Definitionslücken vor, hat die Funktion ganz IR als Definitionsmenge. Dies ist bei echten Polynomen sowie bei den reinen trigonometrischen Funktionen der Fall. Bei »krummen« Werten erfolgt eine Vereinfachung; so hat die Funktion F(x)=1/(x(↑)2-2) nicht etwa eine Definitionslücke bei 1,414 213 562, sondern bei Wurzel 2, weil das mathematisch zwar identisch, aber aussagekräftiger ist. Diese sehr positive und nützliche Eigenschaft gilt auch später für die Anzeige der Nullstellen und der Extremwerte.

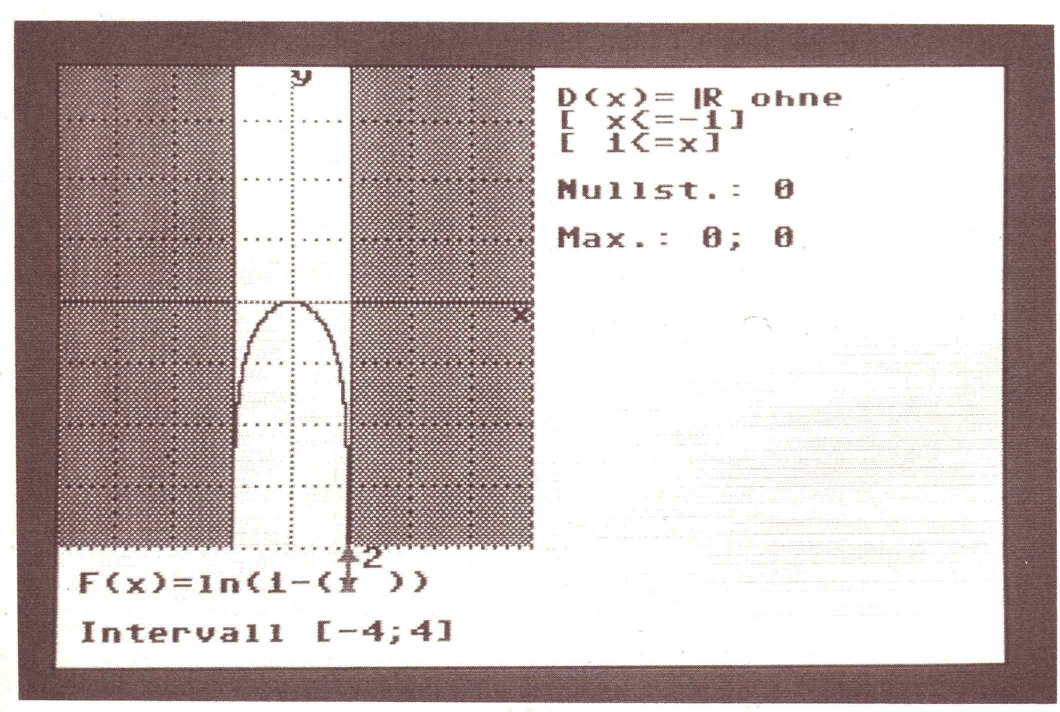
Wie bereits erwähnt, erkennt der Funktionsplotter schon beim Zeichnen des Graphen die Stellen, an denen eine Funktion nicht definiert ist. Die explizite Berechnung ist aber erforderlich, da beim Zeichnen mit gerundeten Werten gearbeitet wird. Für die Berechnung der Nullstellen und Extremwerte benötigt man genaue Werte.

Nachdem der Definitionsbereich errechnet und angezeigt wurde, werden die Nullstellen ermittelt, wenn Sie die entsprechende Frage bejahen. Bei der Nullstellenberechnung wird das Newton-Iterationsverfahren angewandt, das weiter unten im Text genauer erläutert ist. Aufgrund der immer genauer werdenden Näherung kann es (vor allem bei komplexeren Funktionen) einige Minuten dauern, bis alle Ergebnisse feststehen (Abb. 4).

Anschließend stellt das Programm die Frage nach einer Extremwertberechnung. Auch hier antworten Sie wieder mit <J> für »Ja« oder <N> für »Nein«. Da mit einem ähnlichen Algorithmus wie bei der Nullstellensuche gearbeitet wird, sollten Sie sich wieder auf eine gewisse Rechenzeit einstellen. Je komplexer die Formel ist, desto länger dauert die Berechnung. Auch das Intervall hat Einfluß.



[5] Hilfstexte erklären auch falsche Eingaben



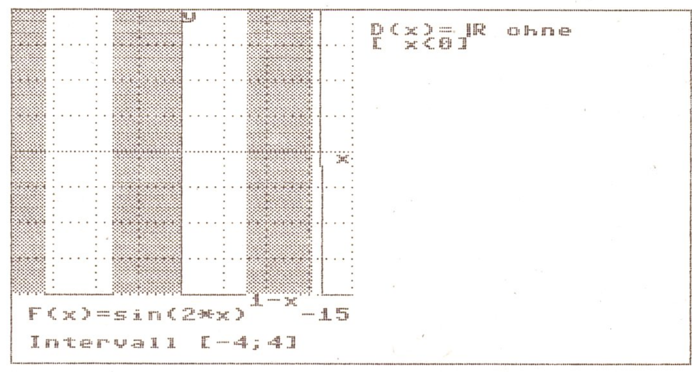
[8] Die nicht definierten Bereiche schraffiert das Programm automatisch

Das Hauptmenü

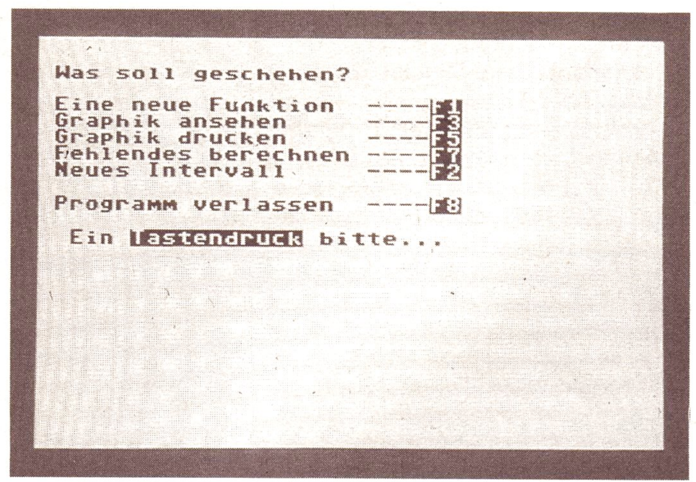
Bei Verneinung oder nach Berechnung und ggf. Ausgabe der Extremwerte befinden Sie sich im Hauptmenü (Abb. 6). Es bietet folgende Optionen, die durch Betätigung der zugeordneten Funktionstaste aktiviert werden:

Neue Funktion

Wirkt wie ein Neustart des Programms, es ist (wie oben beschrieben) eine neue Funktion einzugeben, die danach diskutiert wird.



[7] Die Druckerausgabe ist selbstverständlich



[6] Das Hauptmenü erscheint nach der erstmaligen Berechnung und läßt erweiterte Berechnungen zu

Fehlendes berechnen

Wurde die Funktion noch nicht gezeichnet, können Sie das jetzt nachholen lassen. Ansonsten besteht nachträglich die Möglichkeit zur Berechnung von Definitionsmenge, Nullstellen sowie Extrema. Wurde alles schon berechnet, reagiert der Funktionsplotter nicht auf diesen Menüpunkt.

Variablenliste

a	Hilfsgröße
a$	Hilfsgröße
ad	Adresse
b$	Hilfsgröße
be	Flag für Beträge
d	Hilfsgröße
de	Zähler für errechnete Werte
e$	Funktion im Interpreterformat
en	Intervallgrenze
es$	Funktion wie eingegeben
ex	Flag für Extremwerte
f	Zähler
fa	Faktor für Schrittweite
fe	Flag für Fehlermeldung
fl	Flag
fu	Flag für die Art der Definitionslücke
ge	Hilfsgröße
gp	y-Wert für Grafik
h	Hilfsgröße
h$	Hilfsgröße
h.	Konstanten
hh	Hilfsgröße
hi	Art der Hilfestellung
l	Hilfsgröße
le	Hilfsgröße
ll	Hilfsgröße
lu	Zähler der ausschließenden Definitionslücke
lz	Zähler
m	Hilfsgröße für Extremwerte
me$	Funktion als Basiczeile
nu	Flag für Nullstellen
pl	Flag für Zeichnen
q$	Hilfsgröße
r	Hilfsgröße für Extremwert
s	Zähler
s1	Konstante für sys
s2	Konstante für sys
sw	Schrittweite
v	Hilfsgröße für Grafik
x	Laufvariable
x1	Zwischenspeicher für x
y	Funktionswert
y.	Zwischenspeicher für y
z	Zähler
za	Zähler
ze	Zähler
de()	Zahlenwerte
lu()	Ausschließende Lücken
de$()	Zahlenwerte im String
f$()	Texte

Programm verlassen

Wählen Sie diesen Punkt, wird das Programm ohne Sicherheitsabfrage mit einem Reset verlassen. Der Computer steht dann wieder für andere Aufgaben zur Verfügung.

Neues Intervall

Hier lassen sich das Intervall, in dem die Funktion diskutiert wird, neu festlegen und sodann alle Werte neu ermitteln.

Falls die Funktion gezeichnet wurde, kommen noch zwei weitere Menüpunkte dazu:

Grafik ansehen

Läßt ein Betrachten der Grafik am Bildschirm zu. Ein beliebiger Tastendruck führt zurück ins Menü.

Grafik drucken

Das Programm enthält eine Routine zur Ausgabe des Grafikbildes auf einen Epson LX-800 (Abb. 6 zeigt ein Beispiel). Da die Routine in Basic verfaßt wurde, arbeitet sie relativ langsam (ein komplettes Bild ist knapp neun Minuten in Arbeit). Sie läßt sich dafür ohne weiteres an andere Drucker anpassen. Sollte sich wider Erwarten das Programm nicht an Ihren Drucker anpassen lassen, können Sie einen unserer Treiber anfordern (Näheres s. Textkasten).

Funktionsweise

Damit auftretende Fehler bei der Berechnung der Funktionswerte (wie z.B. ILLEGAL QUANTITY ERROR, DIVISION BY ZERO ERROR, aber auch SYNTAX ERROR) nicht zum Ausstieg führen, wurde der Vektor zur Ausgabe einer Basic-Fehlermeldung auf eine eigene Routine umgelenkt. Sie schreibt bei aufgetretenem Fehler dessen Codenummer in die Speicherzelle 144 (ST) und endet unmittelbar danach mit RTS. Diese Routine wird bei allen Rechenoperationen aktiviert.

Interessant ist vor allem das Unterprogramm zur Berechnung von Definitionslücken ab Programmzeile 980. Nach der Initialisierung der Funktion werden die entsprechenden Felder auf Null gesetzt, einige Vorgabewerte gesetzt und mit der Suche nach funktionsbedingten Lücken begonnen. Diese unterteilen sich in Lücken durch negative Wurzeln, Teilung durch Null, Logarithmen kleiner oder gleich Null und durch nicht ganzzahlige Exponenten (kleiner Null).

Wird das Token für log oder sqr gefunden, wird ein Unterprogramm aufgerufen. In diesem wird das Funktionsargu-

Kurzinfo: Funktionenhilfe

Programmart: math. Anwendungsprogramm
Laden: LOAD"START",8
Start: nach dem Laden RUN eingeben
Steuerung: Menüsteuerung über Tastatur
Besonderheiten: Hilfstexte, Funktion kann in mathematischer Notation eingegeben werden
Länge in Blocks: 133
Programmautor: Michael Suhr

ment als neue Funktion aufgefaßt, und die Nullstellen werden berechnet.

Bei der Suche von Nenner-Nullstellen bei Brüchen wird ähnlich verfahren. Es wird geprüft, ob Stellen mehrfach vorhanden sind und eine Bereichserklärung vorgenommen. Ab-

Extremwertsuche

schließend werden die Werte sortiert, erneut auf mehrfach vorhandene Stellen überprüft, nach Möglichkeit vereinfacht und formatiert. Nach alledem wird die Ursprungsfunktion reinitialisiert, und die Werte werden ausgegeben.

Die Suche nach Nullstellen gestaltet sich folgendermaßen: Zuerst wird die Funktion in ihre Einzelteile zerlegt und diese getrennt getestet. Nach Abschluß der Suche erfolgt eine Plausibilitätsprüfung der Werte und auf Definitionslücken. Dann wird wieder geordnet, vereinfacht, formatiert und ausgegeben.

Newton-Iteration

Um Nullstellen numerisch zu berechnen, benutzt man das Iterationsverfahren nach Newton. Man legt, anschaulich gesprochen, eine Tangente an einen Punkt des Graphen an und ermittelt dann die Nullstelle der sich ergebenden linearen Funktion. Diese ist der neue Startwert. Formal geschrieben lautet dieser Zusammenhang:

```
          F(x0)
x1 = x0 - ------
          F'(x0)
```

Korrekt müßte man statt x0 vielmehr xn schreiben und statt x1: x(n+1).

Der neue Startwert ergibt sich also aus dem alten abzüglich des Differentialquotienten am alten Startwert. Da der Funktionsplotter aus dem Funktionsterm die Ableitung nicht selbst berechnen kann, berechnet er zur Ermittlung des Wertes den Differentialquotienten zweier sehr nahe beieinanderliegender Punkte.

Zur Verdeutlichung mag ein Beispiel hilfreich sein. Die Funktion lautet: F(x)=x(Pfeil nach oben)2-1 und hat die Nullstellen -1 und +1. Zur Newton-Suche beginnen wir mit einem Startwert von 10. Die erste Ableitung dieser Funktion lautet F'(x)=2x, der Differentialquotient an x0=10 hat also den Wert 4,95, somit wird x1=10-4,95=5,09. x2 ist dann 2,6243, x3 ist 1,5026, x4 ist 1,0815 und x5 ist 1,00307. Damit tastet sich der Algorithmus allmählich an die Nullstelle x=1 heran.

Dieses Verfahren findet allerdings nicht mehr als eine Nullstelle. Um möglichst alle zu ermitteln, sucht das Programm mit einer verhältnismäßig kleinen Schrittweite das Intervall durch (deswegen die hohe Rechenzeit) und findet so innerhalb der Auflösung fast alle Nullstellen.

Auch die Extremwertsuche arbeitet nach der Newton-Iteration. Zur Erinnerung: Eine Funktion hat dort einen Extremwert, wo ihre erste Ableitung eine Nullstelle hat. Der Funktionswert der ersten Ableitung gibt die Steigung der Funktion an, somit kann zur Extremwertsuche ein vereinfachtes Iterationsverfahren angewandt werden.

Anstelle der obigen Formel wird hier nun x1 nach

```
          F'(x0)
x1 = x0 - -------
          F''(x0)
```

bestimmt, was für das Programm lediglich mehr Rechenarbeit bedeutet.

Immer wenn ein Extremwert gefunden ist, wird mit Hilfe der zweiten Ableitung (Differentialquotient) geprüft, ob es sich um ein Minimum oder ein Maximum handelt. Zum Schluß werden die Ergebnisse wieder auf Definiertheit geprüft, sortiert, vereinfacht, formatiert und ausgegeben.

Sollten Sie mit dem einen oder anderen mathematischen Fachausdruck noch Probleme gehabt haben, finden Sie im Textkasten ein kleines Mathelexikon, in dem die wichtigsten Begriffe erklärt werden. In der Tabelle finden Sie eine Auflistung der verwendeten Variablen. Ansonsten ist das steuernde Basic-Programm mit REM-Zeilen sehr gut dokumentiert und dadurch der Ablauf nachvollziehbar.

Viel Spaß mit der Funktionenhilfe!

(Nikolaus M. Heusler/gr)

Funktionshilfe und andere Drucker

»Ab Werk« ist das Programm mit einer Druckroutine ausgestattet, die mit Epson-kompatiblen Druckern zusammenarbeitet. Sollten Sie wider Erwarten Schwierigkeiten mit dem Druck haben, steht eine Anpassung zur Verfügung. Sie erhalten diese durch Einsendung einer formatierten Diskette und Rückporto an folgende Adresse:

Markt & Technik Verlag Aktiengesellschaft
Redaktion 64'er Sonderhefte
Druckeranpassung Graph
Hans-Pinsel-Straße 2
8013 Haar bei München

Bitte vergessen Sie nicht, Ihren Druckertyp mitanzugeben.

(gr)

Kleines Mathelexikon

Ableitung: Funktion, die die Steigung einer anderen (der abgeleiteten) Funktion angibt.

Absolutwert: svw. Betrag

Algorithmus: Verfahrensweise, Rechenregel

Argument: Wert, der für die Funktionsvariable x eingesetzt wird.

Betrag: Wert einer Zahl ohne Vorzeichen; der Betrag von -4 ist 4, der Betrag von -23 ist 23, der Betrag von 0 ist 0.

Definitionslücke: Wert, der nicht als Argument einer Funktion eingesetzt werden darf. Die Funktion y=1/x hat etwa die Definitionslücke x=0, da 1/0 nicht definiert ist.

Definitionsmenge: Menge aller Zahlen, die für das Funktionsargument gesetzt werden dürfen.

Differentialquotient: Näherungswert für die Steigung einer Funktion f(x) an der Stelle x0. Man setzt: d(x0) = (x0-(x0+e»/(f(x0)-f(x0+e», wobei e sehr klein ist, und erspart sich so die Berechnung der Ableitung d(x) der Funktion.

Extremwert: Maximum oder Minimum, Nullstelle der ersten Ableitung

Funktion: Vorschrift, die jeder Zahl genau eine andere zuordnet, zum Beispiel y=3*x. x ist das Argument oder die Funktionsvariable, y der Funktionswert.

Funktionsvariable: Argument einer Funktion

Funktionswert: Wert, den eine Funktion annimmt.

Graph: Bildliche Darstellung einer Funktion, des Verlaufs

Intervall: Bereich, Eingrenzung

Iteration: svw. Annäherung. Verfahrensweise, um einen Wert ungefähr zu bestimmen, dessen genaue Berechnung unmöglich ist.

log: dekadischer Logarithmus (Basis 10); Definition: y=log x ermittelt die Zahl y, für die gilt: x=10 hoch y

Maximum: Stelle, an der eine Funktion den höchsten Wert annimmt.

Minimum: Stelle, an der eine Funktion den niedrigsten Wert annimmt.

Nullstelle: Stelle, an der eine Funktion den Wert Null annimmt.

pi: Symbol), wichtige mathematische Konstante mit dem Zahlenwert 3,141 592 654..., wird etwa zur Berechnung in Kreisen verwendet (Kreisfläche = 2*pi*Radius)

Polynom: Term der Form an*x)n+a(n-1)*x)(n-1)+...+a2*x)2+a1*x+a0. n ist der Polynom-Grad.

Tangente: Gerade, die eine Funktion f, an die die Tangente angelegt wird, in einem Punkt P(x0,f(x0» berührt. Die Tangente hat die Steigung der Funktion im Punkt x0, die Steigung hat also den Wert der Ableitung f'(x0).

Term: Rechenausdruck, Formel

(Nikolaus M. Heusler/gr)

Grundlagen/Grafik

Jetzt wird

Wir starten zu einer Reise in die fremdartige Wunderwelt der Grafik des C64. Fast beiläufig lernen Sie den bilderzeugenden Baustein des Computers kennen – den VIC (Video-Interface-Chip).

Zunächst einige allgemeine Informationen über die Bildaufteilung. Der Bildschirm des C64 verfügt im Textmodus über 1000 Positionen und beginnt ab Adresse 1024 ($0400). Sein Ende liegt bei 2023. Jede dieser Adressen kann 8 Bit speichern. Das entspricht einer beliebigen ganzen Zahl zwischen 0 und 255. Der Bildschirmspeicher besteht aus einer Gruppe von 1000 Adressen und wird Farbspeicher oder Farb-RAM genannt. Dieses Farb-RAM belegt den Speicher von 55296 ($D800) bis 56296 ($DBE7). Jede Farb-RAM-Adresse speichert 4 Bit und kann daher eine beliebige ganze Zahl zwischen 0 und 15 aufnehmen.

Die verschiedenen Grafikmodi werden über 47 Steuerregister im Videocontroller-Baustein ausgewählt. Viele Grafikfunktionen lassen sich steuern, indem der richtige Wert über die POKE-Anweisung in eines der Register geschrieben wird, das im Adreßbereich von 53248 ($D000) bis 53294 ($D02E) liegt.

Wahl der Videobank

Der Videocontroller-Baustein kann immer nur auf einen Speicherbereich von 16 KByte zugreifen. Da im C64 bekanntlich 64 KByte RAM untergebracht sind, soll der Videocontroller natürlich auch den ganzen Speicher sehen können. Das ist möglich! Es gibt vier verschiedene Banks (oder Abschnitte), die für jeweils 16 KByte gelten. Es muß lediglich noch geregelt werden, auf welche dieser Abschnitte der Videocontroller-Baustein zugreift. Auf diese Weise kann der Baustein die gesamte Speicherkapazität von 64 KByte ausnutzen. Die Bankanwahl-Bits, die einen Zugriff auf die verschiedenen Speicherabschnitte ermöglichen, befinden sich im Complex-Interface-Adapter #2 (CIA#2) 6526. Über die Basic-Anweisung POKE und PEEK wird eine Bank durch Steuerung der Bits 0 und 1 von Port A des CIA #2 (56576 oder hex $DD00) gewählt. Zur Änderung der Speicherabschnitte müssen diese 2 Bit auf Ausgabe gesetzt sein. Dies wird anhand nachstehender Befehle deutlich:

```
POKE 56578, PEEK (56578) OR 3:REM BITS 0 UND 1 SET-
ZEN
POKE 56576, (PEEK(56576) AND 252) OR A:REM VIDEO-
BANK WECHSELN
```

»A« muß einen der folgenden Werte haben (Tabelle 1). Dieses Konzept der 16-KByte-Abschnitte spielt bei allen Anwendungen des VIC eine Rolle. Sie sollten stets wissen, auf welche Bank der Videocontroller zugreift, denn dadurch wird festgelegt, woher die Zeichendatenmuster kommen und wo sich der Bildschirm sowie der Speicherbereich für die Sprites befinden. Nach dem Einschalten des C64 wird automatisch Bank 0 ($0000 - $3FFF) aktiviert.

WERT VON A	BITS	BANK	START-PLATZ	BEREICH DES VIDEOCONTROLLER BAUSTEINS
0	00	3	49152	($C000-$FFFF)
1	01	2	32768	($8000-$BFFF)
2	10	1	16384	($$000-$7FFF)
3	11	0	0	($0000-$3FFF) (Standardwert)

Tabelle 1. Alle möglichen 16-KByte-Bänke des Videocontrollers

Anmerkung: Der Zeichensatz des C64 ist in den Banks 1 und 3 für den Videocontroller-Baustein nicht verfügbar.

Bildschirmspeicher

Durch POKEn in das Kontrollregister 53272 ($D018 HEX) kann die Adresse des Bildschirmspeichers geändert werden. Dieses Register wird jedoch auch zur Steuerung des jeweils benutzten Zeichensatzes verwendet. Achten Sie daher besonders darauf, diesen Teil des Steuerregisters nicht zu stören. Die oberen 4 Bit steuern den Bildschirmspeicher. Um den Bildschirm in einen anderen Bildbereich zu legen, gibt man folgende Anweisung ein:

```
POKE 53272,(PEEK(53272)AND 15) OR A
```

Hierbei hat A einen der folgenden Werte (Tabelle 2).

Farbspeicher

Der Farbspeicher kann nicht verschoben werden. Er befindet sich stets im Adreßbereich von 55296 ($D800) bis 56295 ($DBE7). Bildschirmspeicher und Farbspeicher werden in den verschiedenen Grafikmodi unterschiedlich benutzt. Ein in einem bestimmten Modus aufgebautes Bild sieht in einem anderen Grafikmodus häufig auch anders aus.

Zeichenspeicher

Für die Programmierung von Grafiken ist es wesentlich, von wo genau der Videocontroller die Zeicheninformation bekommt. Normalerweise erhält der Baustein die Konturen der anzuzeigenden Zeichen vom Character-Generator-ROM. In diesem Baustein sind die Muster gespeichert, die verschiedene Buchstaben, Zeichen, Interpunktionssymbole und alle anderen Zeichen der Tastatur bilden.

Eines der Merkmale des C64 ist seine Fähigkeit, im RAM-Speicher befindliche Muster zu benutzen. Diese RAM-Muster werden vom Anwender selbst kreiert, so daß ein unbegrenzter Satz an Symbolen für Spiele usw. zur Verfügung steht.

Ein normaler Zeichensatz enthält 256 Zeichen, bei dem jedes Zeichen durch 8 Byte bestimmt wird. Da jedes Zeichen also 8 Byte beansprucht, benötigt der komplette Zeichensatz 256 x 8 = 2 KByte. Da der Videocontroller-Baustein gleichzeitig auf 16 KByte zugreift, gibt es acht verschiedene Speicherplatzmöglichkeiten für einen vollständigen Zeichensatz. Sie brauchen natürlich nicht immer einen Zeichensatz zu verwenden. Er muß jedoch stets an einem der acht möglichen Startplätze beginnen. Die Lage des Zeichenspeichers wird durch 3 Bit vom Videocontroller überprüft (53272 oder Hex $D018). Die Bits 3, 2 und 1 steuern die Lage des Zeichensatzes, Bit 0 wird überlesen. Um die Lage des Zeichensatzes zu ändern, benutzen Sie folgende Basic-Anweisung:

```
POKE 53272,(PEEK(53272)AND240) OR A
```






Hierbei hat A einen der folgenden Werte (Tabelle 3).

Greift der VIC auf die Zeichendaten zu, wird das ROM eingeschaltet. Ansonsten wird dieser Bereich von den Ein-/Ausgaberegistern beansprucht, und das Zeichen-ROM kann nur vom VIC erreicht werden. Es kann jedoch passieren, daß Sie das Zeichen-ROM benötigen, und zwar dann, wenn programmierbare Zeichen benutzt werden sollen und eine Kopie eines Teils vom Zeichen-ROM für die Zeichendefinition angelegt wurde. In diesem Fall müssen Sie das Ein-/Ausgaberegister aus- und das Zeichen-ROM einschalten. Jetzt kann kopiert werden. Danach muß das Ein-/Ausgaberegister erneut eingeschaltet werden. Während des Kopierens (bei ausgeschalteter Ein-/Ausgabe) sind keine Unterbrechungen erlaubt. Für Unterbrechungen sind nämlich die Ein-/Ausgaberegister gefragt. Wenn Sie dies vergessen und eine Unterbrechung vornehmen, passiert Unvorhersehbares. Die Tasteneingabe darf während des Kopierens nicht gelesen werden. Um die Tastatur und weitere normale Unterbrechungen abzuschalten, die mit dem C64 möglich sind, benutzen Sie folgende POKE-Anweisung:

A	Bits	Lage	
		dezimal	hexadez
0	0000XXXX	0	$0000
16	0001XXXX	1024	$0400 Std
32	0010XXXX	2048	$0800
48	0011XXXX	3072	$0C00
64	0100XXXX	4096	$1000
80	0101XXXX	5120	$1400
96	0110XXXX	6144	$1800
112	0111XXXX	7168	$1C00
128	1000XXXX	8192	$2000
144	1001XXXX	9216	$2400
160	1010XXXX	10240	$2800
176	1011XXXX	11264	$2C00
192	1100XXXX	12288	$3000
208	1101XXXX	13312	$3400
224	1110XXXX	14336	$3800
240	1111XXXX	15360	$3C00

Tabelle 2. Die oberen vier Bit des Kontrollregisters bestimmen die Lage des Bildschirmspeichers

Wert	Bits	dezimal	hexadezimal
0	XXXX000X	0	$0000-$07FF
2	XXXX001X	2048	$0800-$0FFF
4	XXXX010X	4096	$1000-$17FF ROM-Image in Bank 0 & 2 (Standard)
6	XXXX011X	6144	$1800-$1FFF ROM-Image in Bank 0 & 2
8	XXXX100X	8192	$2000-$27FF
10	XXXX101X	10240	$2800-$2FFF
12	XXXX110X	12288	$3000-$37FF
14	XXXX111X	14336	$3800-$3FFF

Tabelle 3. Die Bits 1, 2, und 3 des Kontrollregisters bestimmen die Lage des Zeichensatzspeichers

```
POKE 56334,PEEK(56334)AND 24 (Interrupt AUS)
```

Wenn Sie den Zugriff auf das Zeichen-ROM beendet haben, wird die Tastatur durch folgende POKE-Anweisung eingeschaltet:

```
POKE 56334,PEEK(56334)OR1 (Interruppt EIN)
```

Durch folgende POKE-Anweisung wird die Ein-/Ausgabe aus- und das Zeichen-ROM eingeschaltet:

```
POKE 1, PEEK (1)AND 251
```

Das Zeichen-ROM befindet sich nun im Bereich von 53248 bis 57343 ($D000 bis $DFFF). Um die Ein-/Ausgabe für den normalen Betrieb zu aktivieren, ist folgende POKE-Anweisung erforderlich:

```
POKE 1,PEEK(1)OR4
```

Standardzeichenmodus

Beim Einschalten des C64 befindet sich dieser im Standardzeichenmodus, in dem Sie normalerweise programmieren. Zeichen können aus dem ROM oder dem RAM gelesen werden. Normalerweise wird jedoch auf die Zeichen im ROM zugegriffen. Benötigen Sie für ein Programm spezielle Grafikzeichen, so sind lediglich die neuen Zeichenmuster im RAM zu definieren und der VIC anzuweisen, die Zeicheninformationen aus dem RAM und nicht aus dem Zeichen-ROM zu nehmen.

Um Zeichen auf dem Bildschirm in Farbe anzuzeigen, greift der Videocontroller-Baustein auf den Bildschirmspeicher zu, damit der Zeichencode für diesen Bildschirmplatz bestimmt wird. Gleichzeitig greift er auf den Farbspeicher zu, um die Farbe für die Zeichenanzeige festzulegen. Der Zeichencode wird vom Videocontroller in die Startadresse des 8-KByte-Satzes mit Ihrem Zeichenmuster umgesetzt.

Die Transformation ist nicht zu kompliziert, zur Erstellung der gewünschten Adresse werden jedoch verschiedene Punkte kombiniert. Zunächst ist der von Ihnen bei der POKE-Anweisung für den Bildschirmspeicher benutzte Zeichencode mit 8 zu multiplizieren. Danach wird der Anfang vom Zeichenspeicher addiert. Nun werden die Bankanwahl-Bytes berücksichtigt. Hierzu wird die Basisadresse addiert. Anhand der folgenden einfachen Gleichung können Sie sehen, wie dies gemeint ist:

Zeichenadresse = Bildschirmcode x 8 + (Zeichensatz x 2048) + (Bank x 16348)

Zeichendefinition

Jedes Zeichen wird aus einer Matrix von 8 x 8 Punkten gebildet. Hierbei können die einzelnen Punkte entweder ein- oder ausgeschaltet sein. Beim Commodore 64 sind die Zeichen-Bits im Zeichengenerator ROM abgelegt. Jedes Zeichen ist dabei als Satz von 8 Byte gespeichert. Jedes Byte steht für das Punktmuster einer Reihe im Zeichen und jedes Bit für einen Punkt. Der Zeichenspeicher im ROM beginnt ab 53248 (bei ausgeschalteter Ein-/Ausgabe). Die ersten 8 Byte ab 53248 ($D000) bis 53255 ($D007) enthalten das Muster für das Zeichen @, dessen Zeichencodewert im Bildschirmspei-

cher 0 ist. Die nächsten 8 Byte ab 53265 ($D00F) enthalten die Information zur Bildung des Buchstabens A (Abb. 1).

Jeder vollständige Zeichensatz beansprucht eine Speicherkapazität von 2 KByte (2048 Bit). Insgesamt sind 256 Zeichen vorhanden. Da es insgesamt zwei Zeichensätze gibt, enthält der ROM-Zeichenspeicher also 4 KByte Speicherplatz.

Programmierbare Zeichen

Die Zeichen sind im ROM gespeichert; darum sieht es so aus, als ob sie für frei programmierbare Zeichen nicht geändert werden könnten. Der Speicherplatz, der dem VIC mitteilt, wo die Zeichen zu finden sind, ist jedoch ein programmierbares Register. Dieses kann so geändert werden, daß es auf viele Speicherbereiche zeigt - indem der Zeichenspeicherzeiger so geändert wird, daß er auf das RAM zeigt. Alle Buchstaben, Zahlen oder Standardgrafikzeichen vom C64 müssen zuerst in den RAM-Speicher kopiert werden, damit Sie ihn in Ihrem Programm benutzen können.

Achtung: Achten Sie darauf, daß Ihr Zeichensatz nicht vom Basic-Programm, das auch das RAM benutzt, überschrieben wird!

Zwei Adressen im C64 dürfen nicht als Beginn des Zeichensatzes gewählt werden: Adresse 0 und Adresse 2048. Der erste nicht, weil das System auf Seite 0 (0-Page) wichtige Daten speichert. Adresse 2048 ist der Beginn Ihres Basic-Programms.

Für Ihren Zeichensatz stehen jedoch noch sechs weitere Anfangspositionen zur Verfügung. Am besten wählen Sie hierzu den Bereich ab 12288 ($3000), indem Sie zu den unteren 4 Bit in Speicherzelle 53272 den Wert 12 addieren. Probieren Sie nun folgende POKE-Anweisung aus:

```
POKE 53272,(PEEK(53272)AND240)+12
```

Sofort sind alle Buchstaben vom Bildschirm verschwunden, weil nämlich bis jetzt noch kein Zeichensatz ab Adresse 12288 steht...nur zufällige Bytes. Kehren Sie durch Betätigen der Tasten <RUN/STOP + RESTORE> wieder zurück in den Normalmodus.

Nun wollen wir Grafikzeichen konstruieren. Um Ihren Zeichensatz zu schützen, sollten Sie die Speicherkapazität für das Basic reduzieren. Der Speicher in Ihrem Computer bleibt unverändert. Sie haben lediglich dem Basic die Anweisung gegeben, einen bestimmten Teil nicht zu benutzen. Tippen Sie ein:

```
PRINT FRE(0)-(SGN(FRE(0))<0)*65535
```

Die angezeigte Zahl gibt die unbenutzte Speicherkapazität an. Geben Sie nun folgendes ein:

```
POKE 52,48:POKE 56,48:CLR
```

Und nun:

```
PRINT FRE (0)-(SGN(FRE(0))<0)*65535
```

Sehen Sie die Änderung? Das Basic nimmt nun an, es stehe weniger Speicherkapazität zur Verfügung. In diesem gewonnenen Speicherplatz können Sie nun Ihren Zeichensatz unterbringen. Als nächstes sind die Zeichen im RAM zu definieren. Zu Beginn stehen ab 12288 ($3000) zufällige Daten. Durch folgendes Programm werden 64 Zeichen vom ROM- in den RAM-Zeichensatz übertragen:

```
5 PRINT CHR$(142)
10 POKE 52,48:POKE 56,48:CLR
20 POKE 56334,PEEK (56334) AND 254
30 POKE 1,PEEK (1) AND 251
40 FOR I = 0 TO 511:POKE I +12288,
   PEEK(I+53248):NEXT
50 POKE 1,PEEK(1)OR4
60 POKE 56334,PEEK(56334)OR1
70 END
```

Geben Sie nun ein:

```
POKE 53272,(PEEK(53272)AND240)+12
```

Nichts passiert, stimmt's? Fast nichts! Der C64 bekommt die Zeicheninformationen nun vom RAM, und nicht vom ROM. Da wir jedoch die Zeichen genau aus dem ROM kopiert haben, ist kein Unterschied zu sehen ...noch nicht. Die Zeichen können nun leicht geändert werden. Löschen Sie den Bildschirm und drücken Sie die Taste <@>. Bewegen Sie den Cursor um einige Zeilen nach unten und geben Sie dann ein:

```
FOR I = 12288 TO 12288 + 7:POKE I,255-PEEK(I):NEXT
```

Sie haben soeben ein <@> in Reversdarstellung erstellt.

Bewegen Sie nun den Cursor wieder zum Programmanfang und drücken Sie die Taste <RETURN> erneut, um das Zeichen noch einmal umzukehren. Die Tabelle der Bildschirm-Codes zeigt Ihnen, wo die einzelnen Zeichen im RAM stehen. Denken Sie daran, daß zur Speicherung jedes Zeichens acht Speicherplätze benötigt werden. Wird ein anderes Zeichen gewünscht, so ist vorher noch etwas zu berücksichtigen. Was ist zu tun, wenn Sie nun Zeichennummer 154, ein umgekehrtes Z, wünschen? Sie können das erreichen, indem Sie ein Z umkehren, oder Sie können den Satz der umgekehrten Zeichen aus dem ROM kopieren oder einfach das eine Zeichen aus dem ROM holen und ein nicht benötigtes Zeichen im RAM dadurch ersetzen. Nehmen wir an, Sie benötigen das Zeichen > nicht mehr. Es soll also gegen das negativ dargestellte Zeichen Z ausgetauscht werden. Geben Sie folgendes ein:

```
FOR I = OTO 7:POKE 12784+I,255-PEEK(I+12469):NEXT
```

(Diese Änderung betrifft jedoch nur die Darstellung auf dem Bildschirm. Auch wenn das Zeichen wie ein umgekehr-

[1] Der Buchstabe »A« steht im Zeichen-ROM in den Speicherstellen von 53256 bis 53263

Belegung	binär	PEEK
	00011000	24
	00111100	60
	01100110	102
	01111110	126
	01100110	102
	01100110	102
	01100110	102
	00000000	0

[2] Die Bits des Buchstabens »A« im Normalmodus

Belegung	Bit-Muster
**	00011000
****	00111100
** **	01100110
******	01111110
** **	01100110
** **	01100110
** **	01100110
	00000000

[3] Die Bits des Buchstabens »A« im Multicolor-Modus

Darstellung	Bit-Muster
AABB	00011000
CCCC	00111100
AABBAABB	01100110
AACCCCBB	01111110
AABBAABB	01100110
AABBAABB	01100110
AABBAABB	01100110
	00000000

tes Z aussieht, wirkt es in einem Programm doch immer noch als >): Probieren Sie das an einem Beispiel aus, bei dem dieses Zeichen benötigt wird.

Fassen wir zusammen: Sie können nun Zeichen aus dem ROM ins RAM kopieren. Hinsichtlich der programmierbaren Zeichen fehlt Ihnen also nur noch ein Punkt (und zwar der beste!) ....die Kreation Ihrer eigenen Zeichen. Wissen Sie noch, wie Zeichen im ROM gespeichert sind? Die Bit-Muster der Zeichen-Bytes geben direkt das Zeichen wieder. Werden 8 Byte übereinander angeordnet und jedes Byte als achtstellige Binärzahl geschrieben, entsteht eine 8 x 8-Matrix, die das Zeichen darstellt. Enthält ein Bit eine 1, so ist an diesem Platz ein Punkt; enthält es eine 0, ist an diesem Platz eine Leerstelle.

Mehrfarbige Grafiken

Durch die standardmäßige hochauflösende Grafik können Sie selbst Einzelpunkte auf dem Bildschirm ansteuern. Für jeden Punkt im Zeichensatz stehen zwei Werte zur Verfügung: 1 für EIN und 0 für AUS. Hat ein Punkt den Wert 1, wird er in der von Ihnen für die jeweilige Bildschirmposition gewählten Farbe angezeigt. Bei der hochauflösenden Grafik können alle Punkte innerhalb der 8 x 8-Matrix entweder in der Hinter- oder Vordergrundfarbe angezeigt werden. Hierdurch wird die Farbauflösung innerhalb dieses Bereichs eingeschränkt. So können z.B. Schwierigkeiten entstehen, wenn sich zwei Linien mit verschiedenen Farben kreuzen.

Dieses Problem wird durch den Mehrfarbenmodus gelöst. Hierbei kann jeder Punkt eine von vier Farben annehmen: Bildschirmfarbe (Hintergrundfarbregister #0), die Farbe im Hintergrund #1, die Farbe im Hintergrundfenster #2 oder die Zeichenfarbe. Die einzige Einschränkung liegt in der horizontalen Auflösung, da im Mehrfarbenmodus jeder Punkt doppelt so breit ist wie bei Hochauflösung. Es überwiegen jedoch bei weitem die vielen Vorteile des Mehrfarbenmodus.

Zum Einschalten des Modus für mehrfarbige Zeichen wird Bit 4 des Steuerregisters durch folgende POKE-Anweisung bei 53270 ($D016) gesetzt.:

```
POKE 53270,PEEK(53270)OR 16
```

Zum Abschalten dieser Betriebsart wird Bit 4 der gleichen Speicherzelle (53270) durch nachstehende POKE-Anweisung auf 0 gesetzt:

```
POKE 53270,PEEK(53270)AND239
```

Dieser Mehrfarbenmodus wird für jede Bildschirmstelle ein- oder ausgeschaltet, so daß Mehrfarbengrafiken und Grafiken mit hoher Auflösung (Hires) kombiniert werden können. Dies wird über Bit 3 im Farbspeicher gesteuert. Der Farbspeicher beginnt ab 55296 ($D800). Ist die Zahl im Farbspeicher kleiner als 8 (0-7), so gilt für die entsprechende Stelle auf dem Bildschirm Hochauflösung in der gewählten Farbe (0-7). Ist die Zahl im Farbspeicher größer oder gleich 8 (8-15), dann wird die entsprechende Stelle im Mehrfarbenmodus angezeigt.

Die Zeichenfarbe einer Bildschirmposition kann durch eine POKE-Anweisung im Farbspeicher geändert werden. Durch das POKEn einer Zahl von 0 bis 7 werden die Zeichen in normaler Farbdarstellung angezeigt. Durch das POKEn einer Zahl zwischen 8 und 15 gilt für die entsprechende Bildschirmstelle der Mehrfarbenmodus. Das heißt, durch Einschalten von Bit 3 im Farbspeicher wird der Mehrfarbenmodus und durch Ausschalten der normale Hochauflösungsmodus gewählt.

Gilt für eine Bildschirmstelle der Mehrfarbenbetrieb, wird durch die Zeichen-Bits bestimmt, welche Farben für die Punkte angezeigt werden. Nachstehend sehen Sie die Darstellung des Buchstabens A und das entsprechende Bit-Muster (Abb. 2).

Im normalen oder Hires-Modus wird die Bildschirmfarbe bei jedem 0-Bit und die Zeichenfarbe stets da angezeigt, wo das Bit 1 ist. Beim Mehrfarbenmodus werden die Bits paarweise benutzt (Abb. 3).

BIT-PAAR	FARBREGISTER	SPEICHERPLATZ
00	HINTERFGRUNDFARBE # 0 (BILDSCHIRMFARBE)	53281 ($D021)
01	HINTERGRUNDFARBE # 1	53282 ($D022)
10	HINTERGRUNDFARBE # 2	53283 ($D023)
11	DURCH DIE UNTEREN 3 BITS IM FARBSPEICHER BESTIMMTE FARBE	FARBSPEICHER

Tabelle 4. Mögliche Farbkombinationen für Multicolor-Zeichen

ZEICHENCODE			HINTERGRUNDFARBREGISTER	
BEREICH	BIT 7	BIT 6	NUMMER	ADRESSE
0 - 63	0	0	0	53281($D021)
64 - 127	0	1	1	53282($D022)
128 - 191	1	0	2	53283($D023)
192 - 255	1	1	3	53284($D024)

Tabelle 5. Im Hintergrund-Farbmodus können die ersten 64 Zeichen mit unterschiedlichen Hintergrundfarben dargestellt werden

Im Bildbereich werden die durch AA gekennzeichneten Stellen in der Hintergrundfarbe #1, die durch BB gekennzeichneten Stellen in der Hintergrundfarbe #2, und die durch CC gekennzeichneten Stellen in der Zeichenfarbe dargestellt. Dies wird entsprechend nachstehender Tabelle 4 durch Bit-Paare bestimmt.

Hintergrund-Farbmodus

In diesem Modus können Sie für jedes einzelne Zeichen sowohl die Hintergrund- als auch die Vordergrundfarbe steuern. So ist es z.B. möglich, auf einem weißen Bildschirm ein blaues Zeichen mit gelbem Hintergrund anzuzeigen. Für den erweiterten Farbmodus stehen vier Register zur Verfügung. Für jedes kann eine der 16 Farben gewählt werden.

In diesem Modus wird über den Farbspeicher die Vordergrundfarbe festgelegt. Die Anwendung ist die gleiche wie beim Standardzeichenmodus.

Bei erweitertem Modus ist die Anzahl der verschiedenen anzeigbaren Zeichen jedoch eingeschränkt. Ist der erweiterte Farbmodus eingeschaltet, können nur die ersten 64 Zeichen des Zeichen-ROM (oder die ersten 64 in Ihrem programmierbaren Zeichensatz) benutzt werden. Zwei Bit des Zeichencodes werden für die Wahl der Hintergrundfarbe benutzt.

Der Zeichencode (die auf dem Bildschirm gePOKEte Zahl) vom Buchstaben A ist eine 1. Im erweiterten Farbmodus erscheint nach dem POKEn einer 1 ein A. Normalerweise muß nach dem POKEn von 65 das Zeichen mit dem Zeichencode (CHR$) 129, also ein invertiertes A, erscheinen. Dies passiert nicht im erweiterten Farbmodus. Es erscheint genau dasselbe A wie vorher, jedoch eine andere Hintergrundfarbe. Entnehmen Sie die Codes der Tabelle 5.

Zum Einschalten des erweiterten Farbmodus wird Bit 6 des VIC-II-Registers mit der Adresse 53265 ($D001) auf 1 gesetzt. Dies geschieht durch folgende POKE-Anweisung:

```
POKE 53265,PEEK(53265)OR 64
```

Zum Ausschalten wird Bit 6 des VIC-II-Registers mit der Adresse 53265 ($D011) auf 0 gesetzt:

```
POKE 53265,PEEK(53265)AND 191
```

Damit sind wir am Schluß des Kurses angelangt. (gr)

Gekürzter Auszug aus dem Buch »Alles über den C 64« aus der Commodore-Sachbuchreihe, Markt & Technik Verlag, Hans-Pinsel-Str. 2, 8013 Haar, Best.-Nr. ISBN 3-89090-379-7.

Machen Sie mit!

Wollen Sie mithelfen, die folgenden 64'er-Sonderhefte optimal nach Ihren Wünschen zu gestalten?

Dann bitten wir Sie, die folgenden Fragen kritisch zu beantworten, die Seite aus dem Heft zu trennen und sie in einem frankierten Briefumschlag an folgende Adresse zu schicken (Drucksache genügt):

Markt & Technik Verlag AG
Redaktion 64'er-Sonderhefte
Stichwort: Mitmachaktion
Hans-Pinsel-Str. 2
8013 Haar b. München

Als Bonbon verlosen wir 10 Sonderhefte mit Diskette.
Wenn ich gewinne, wünsche ich mir Sonderheft Nr. ______

Wie alt sind Sie? __________ Jahre

Welche Artikel fanden Sie in dieser Ausgabe besonders gut (Note 1) oder schlecht (Note 6):

Artikel	Seite	Note (1 bis 6)
1.		
2.		
3.		
4.		
5.		
6.		

In den nächsten 64'er-Sonderheften wünsche ich mir folgende Themen:

Thema	ja	nein
Spiele	O	O
Adventures	O	O
Grafik	O	O
Anwendungen	O	O
Tips & Tricks	O	O
Hardware basteln	O	O
Basic	O	O
Assembler	O	O
Floppy	O	O
Drucker	O	O
Sound	O	O
Geos	O	O
C 128	O	O
Einsteiger	O	O

Die Artikel, Beiträge und Programmbeschreibungen sind für mich:

Bewertung	ja	manchmal	nein
verständlich	O	O	O
ausführlich genug	O	O	O
gut gemischt	O	O	O

Ich möchte an der redaktionellen Gestaltung mitarbeiten. Meine Vorschläge:

____
____
____
____
____

Folgende eigene Programmentwicklungen kann ich zur Veröffentlichung anbieten:

____
____
____
____
____

Zu folgenden interessanten Anwendungen des C64/C128 biete ich ausführliche Berichte oder Artikel an:

____
____
____
____
____

Ich stehe mit dem C64 vor folgenden Problemen:

____
____
____
____

Die neue POWER PLAY ist da !

"Auf der Consumer Electronics Show in Las Vegas gab´s jede Menge Neuheiten für alle Systeme."

Spiele satt

Der ausführliche Messebericht informiert Euch über aktuelle News und Trends aus Amerika. Neben tonnenweise neuen Modulen für sämtliche Videospiele darf man besonders auf Wing Commander 2 von Origin gespannt sein.



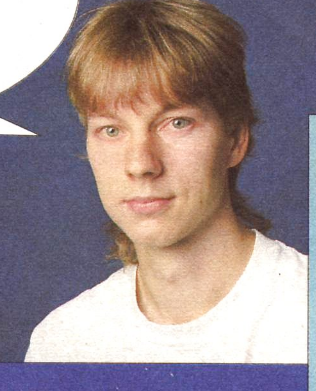
"Wer ist wer im Land der Drachen? Wir nehmen die interessantesten Drachenabkömmlinge in Computer- und Videospielen unter die Lupe."



Drachenpracht

Keine Tieratt begegnet uns so oft auf dem Bildschirm wie die feurigen Flammenfaucher. Ob Action- oder Rollenspiel: ohne Schuppentier geht kaum etwas. Aus diesem Grund haben wir einige Exemplare dieser Rasse genauer betrachtet und viele interessante Details herausgefunden.

"Welche Module sollte sich jeder Game Boy-Besitzer besorgen? Wir stellen die fünfzehn besten Spiele vor."



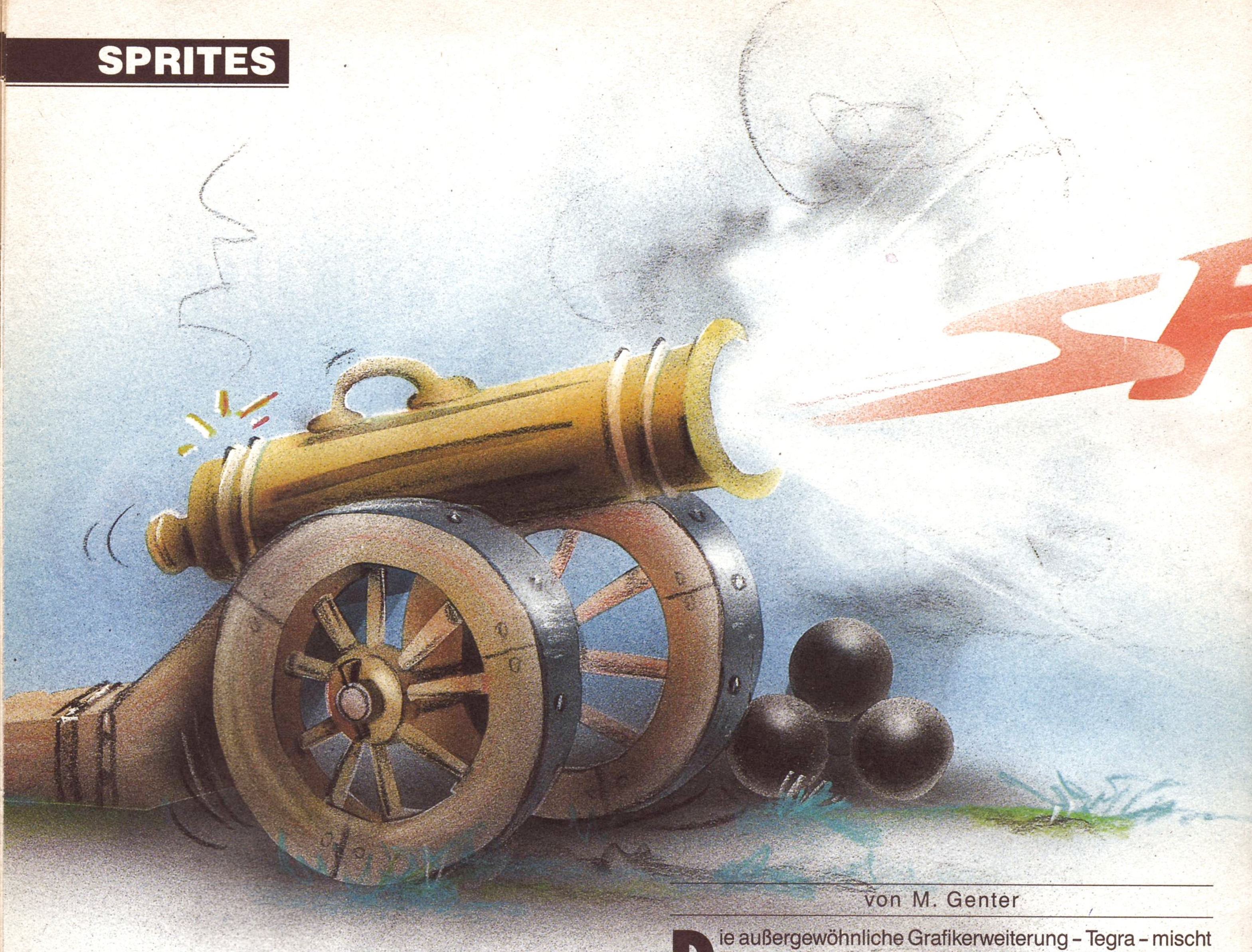
Game Boy-News

Der Game Boy war in Deutschland der Weihnachtsrenner. Knapp eine halbe Million Stück wurden bis heute verkauft. Wir präsentieren neues, interessantes Zubehör und empfehlen 15 Module, die jeder haben sollte.





Holt Euch POWER PLAY jetzt bei Eurem Händler !



von M. Genter

Wie wäre es denn mit Text und Grafik gemischt auf einem Bildschirm? Nicht wie üblich eine Hälfte Grafik, die andere Text, sondern übereinander. Dazu noch eine Befehlserweiterung, mit der sich auch Diagramme komfortabel konstruieren lassen.

Tegra - Text und Grafik in einem

NICHTS IST UNMÖGLICH

Die außergewöhnliche Grafikerweiterung – Tegra – mischt Text und Grafik und stellt Ihnen 23 neue Basic-Befehle zur Verfügung. Um sowohl Grafik als auch Text unabhängig voneinander darzustellen, wurde ein besonderer Kniff gewählt:'

Die Grafik wird aus sieben Sprites erzeugt, die fünfmal jeweils untereinander auf dem Bildschirm erscheinen. Dadurch ergibt sich eine Auflösung von 160 x 100 Punkten. Nur durch diesen Trick lassen sich unabhängig voneinander Grafik und Text darstellen. Zusätzlich sind zwei Tegra-Erweiterungen mit auf Diskette:

1. Ein Grafikkonverter wandelt Hires-Bilder in das Tegra-Format um.
2. Das komfortable Malprogramm Hi-Tegra besitzt Befehle, die Sie an das legendäre Malprogramm Hi-Eddi erinnern.

Bevor Sie eines der beiden Hilfsprogramme verwenden können, laden Sie mit folgender Eingabe die eigentliche Befehlserweiterung:

```
LOAD"TEGRA",8
```

Gestartet wird mit RUN. Nach der Initialisierung werden die Befehle am Bildschirm aufgelistet (Abb. 1). Sie können im Direkt- und im Programmodus verwendet werden. Alle Befehle sind als Basic-Tokens integriert und können abgekürzt werden. Für die einzelnen Parameter lassen sich definierte oder berechnete Variablen verwenden. Dazu ein Beispiel:

```
100 A=10:B=25
110 DOT A,B
```

Der Befehl »DOT« wird weiter unten erklärt.

Achtung

Nach einer IF-THEN muß ein Doppelpunkt stehen. Dazu ein Beispiel:

```
100 IF A=0 THEN:MIX
```



Vor Lade- und Speicheroperationen mit der Floppy ist die Grafik grundsätzlich mit »NOMIX« auszuschalten. Der Grund liegt im aufwendigen Timing der Datenübertragung von und zur Diskettenstation: Sind mehr als drei Sprites eingeschaltet, hängt sich das Betriebssystem auf – der Computer läßt sich nur durch RESET oder <RUN/STOP RESTORE> zurücksetzen.

Die folgenden Befehle stehen zusätzlich zum Basic des C64 zur Verfügung (Abb. 2):

MIX

Schaltet das Mischen von Text und Grafik ein. Nach dem Programmstart befinden sich willkürliche Daten im Grafikspeicher. Verwenden Sie darum **vor** dem ersten »MIX« den Befehl »CLEAR« (s. u.).

NOMIX

Schaltet das Mischen wieder aus.

Achtung:

Bevor Sie Lade- oder Speicheroperationen durchführen, muß dieser Befehl verwendet werden.

CLEAR

Löscht den Grafikspeicher. Der Textbildschirm wird nicht verändert.

DOT x,y(,modus)

Damit kann man Punkte der Grafik setzen, löschen und invertieren.

»x« kann die Werte von 0 bis 159, »y« Werte von 0 bis 99 annehmen. Der Punkt 0/0 liegt in der Ecke links oben.

Mit »Modus« wird die Darstellungsform der Grafik bestimmt:

0 = Punkt löschen (z.B. DOT 10,20,0)
1 = Punkt setzen (z.B. DOT 10,20,1)
2 = Punkt invertieren (z.B. DOT 10,20,2)

Wird der Modus weggelassen, gilt der Punkt als gesetzt (z.B. DOT 10,20).

```
 **** TEGRA V4.0 BY M. GENTNER ****
 64K RAM SYSTEM  36863 BASIC BYTES FREE
KILL      MIX       NOMIX     CLEAR
INVERS    PRIORITY  COLOR@    COLOR
PUT       TAKE      DOT       PIXEL
LINE      BOX       CIRCLE    CHAR
BLOCK     SPRITE    SET       SIZE
HARDCOPY  INFO      SPRCOL    FILL

READY.
```

[1] Nach dem Einschalten erhalten Sie eine Befehlsübersicht

PIXEL (x,y)

Hiermit läßt sich ein Grafikpunkt abfragen. Mögliche Ergebnisse sind

0 = gelöscht
1 = gesetzt

Dazu wieder ein Beispiel: PRINT PIXEL (1,50)

LINE x1,y1,x2,y2(,modus)

Dieser Befehl zieht eine Linie vom Punkt x1/y1 zum Punkt x2/y2. Für die Parameter (Modus) gilt das gleiche wie bei DOT.

```
LINE 0,99,159,0
```

zeichnet eine Linie von rechts oben nach links unten.

```
LINE 0,99,159,0,0
```

löscht diese Linie wieder.

BOX x1,y1,x2,y2(,modus)

Box zeichnet ein Rechteck mit den Eckpunkten x1/y1, x2/y1, x2/y1, x1/y2 und x2/y2.

CIRCLE xmitte, ymitte, xradius, yradius(,modus)

Hiermit lassen sich Ellipsen (und Kreise) mit dem Mittelpunkt xmitte/ymitte und den beiden Radien xradius und yradius zeichnen. Für einen Kreis müssen beide Radien gleich sein. Die Parameter werden so überprüft, daß keine Teile der Figur außerhalb des Bildschirms liegen können.

CHAR string,x,y(,modus)

Dient dem Schreiben von Text direkt in die Grafik. Dafür werden die Zeichendaten aus dem Zeichen-ROM direkt in den Grafikspeicher übertragen. Die Schrift erscheint viermal so groß (ein Grafikpunkt ist viermal so groß wie normal). Der Befehl eignet sich dadurch gut für Überschriften. x und y geben den linken oberen Eckpunkt des Textes an; x darf zwischen 0 und 152 liegen, y zwischen 0 und 93.

FILL x,y

Zum Füllen von beliebigen Flächen um den Punkt x/y.

PRIORITY pbit

Mit diesem Befehl bestimmt man, ob die Grafik über dem Text (PRIORITY 0) oder der Text über der Grafik (PRIORITY 1) liegen soll.

INVERS

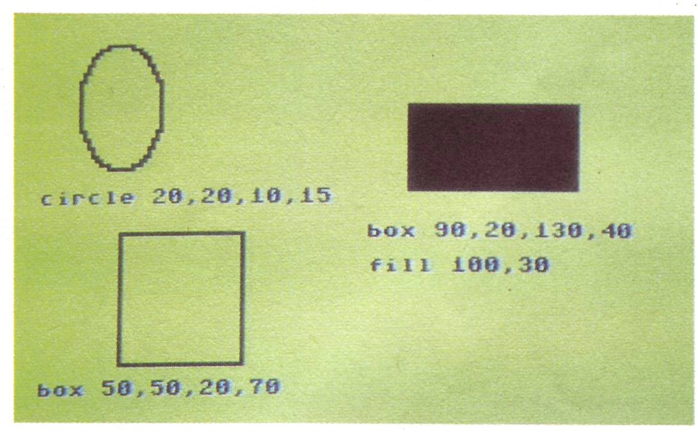
Invertiert die gesamte Grafik.

COLOR farbe

Dieser Befehl setzt die Farbe der gesamten Grafik. Der Parameter muß zwischen 0 und 15 liegen. Die Farbcodes sind identisch mit den im Handbuch beschriebenen.

COLOR@x,y,farbe

Jeder Sprite-Block der Grafik kann anders gefärbt werden, also jeder 24 x 21-Punkte-Block eine andere Farbe annehmen. x und y geben hier x- und y-Koordinate des gewünschten Blocks an (x von 0 bis 6 und y von 0 bis 4).



[2] Einige Beispiele für Texteingaben

PUT typ, filename + ",P,W"
Speichert eine Grafik auf Diskette (inklusive Farbspeicher). Bedeutung des Typ-Parameters:
1 = Grafik speichern
2 = Textbildschirm speichern
3 = Beides auf einmal speichern
Der Typ-Parameter wird mitgespeichert.

TAKE filename
Liest ein mit PUT gespeichertes File von Diskette. Da der Typ-Parameter gespeichert ist, muß er nicht angegeben werden.

Achtung: Bei Put und Take wird die Grafik automatisch abgeschaltet.

TEGRA-Befehlstabelle

Befehl	Aktion
MIX	Schaltet Mischen ein
NOMIX	Schaltet Mischen aus
CLEAR	Löscht Grafikspeicher
DOT	Punkte setzen
PIXEL	Abfrage nach Punkt
LINE	Linien ziehen
BOX	Rechteck
CIRCLE	Elypsen und Kreise
CHAR	Text schreiben
FILL	Füllen von Flächen
PRIORITY	Text über (1) oder unter (0) Grafik
INVERS	invertiert Grafik
COLOR	Farbe für gesamte Grafik
COLOR@	färbt Spriteblock
PUT	Speichern
TAKE	Laden
HARDCOPY	Hardcopy (MPS801)
SPRITE	schaltet freies Sprite ein
SPRCOL	setzt Farbe von Sprite
BLOCK	setzt Blockpointer
SIZE	Vergrößerung Sprite
SET	Position des Sprites
INFO	Befehlstabelle ausgeben
KILL	Reset

Kurzinfo: Tegra

Programmart: Grafikerweiterung
Laden: LOAD"TEGRA",8
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur
Besonderheiten: Text und Grafik mischbar
Benötigte Blocks: 19
Programmautor: M. Genter

HARDCOPY typ
Erzeugt eine Hardcopy auf einem MPS-801-kompatiblen Drucker. Folgende Variationen stehen zur Verfügung:
0 = Grafik-Hardcopy im Miniformat
1 = Grafik-Hardcopy in normaler Größe
2 = Text-Hardcopy
3 = Grafik + Text-Hardcopy (mit OR gemischt)
4 = Grafik + Text-Hardcopy (mit EXOR gemischt)
Da für die Grafikerzeugung nur sieben der acht Sprites verwendet werden, ist das achte frei verwendbar. Dafür existieren fünf Befehle:

SPRITE sbit
SPRITE 1 schaltet das freie Sprite ein, SPRITE 0 schaltet es aus.

SPRCOL farbe
Setzt die Farbe des Sprite.

BLOCK spriteblock
Setzt den Block-Pointer (z.B. BLOCK 11).

SIZE xbit, ybit
Dieser Befehl steuert die x- und y-Vergrößerung des Sprite.
0 = normal
1 = doppelt so hoch/breit
Dazu ein Beispiel:

```
SIZE 1,0
```

setzt die x- und löscht die y-Vergrößerung.

SET x,y
Setzt die Position des Sprites; x kann im Bereich von 0 bis 511, y im Bereich von 0 bis 255 liegen.

Zusätzlich zur Grafik stehen zwei allgemeine Befehle zur Verfügung:

INFO
Gibt die Tegra-Befehlstabelle am Bildschirm aus.

KILL
Schaltet Tegra ab (Reset).

Kurzinfo: Hi-Tegra

Programmart: Malprogramm für Tegra
Laden: LOAD"HI-TEGRA",8
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur und Joystick in Port 2
Besonderheiten: Text und Grafik unabhängig voneinander zu bearbeiten
Benötigte Blocks: 11
Programmautor: M. Genter

HI-TEGRA:
Hi-Tegra ist ein Malprogramm und läßt sich mit dem Joystick in Port 2 steuern. Die Bedienung ähnelt der von Hi-Eddi. Geladen wird (bei aktiviertem Tegra) mit:

```
LOAD "HI-TEGRA",8
```

und gestartet wird mit RUN.

Es erscheint ein flackernder Cursor, der sich über den Joystick bewegen läßt. Folgende Tastenfunktionen sind aktiv:
Cursor-Tasten – Bewegung des Cursors in 4-Punkte-Schritten.
<**SHIFT HOME**> – Löschen der gesamten Grafik
<**1**> – Invertieren der Grafik
<**F2**>,<**F4**>,<**F6**>,<**F8**> – Setzt einen der vier Tabulatoren.



[3] Eine grafische Weltkarte und Text gemischt

<**F1**>,<**F3**>,<**F5**>,<**F7**> – Setzt den Cursor an eine der Tabulatorpositionen.

<**D**> – Schaltet auf Draw-Modus.

In diesem Modus gilt:

Feuerknopf = Punkt setzen

<**SHIFT**> + Feuer = Punkt löschen

<**L**> – Schaltet auf Line-Modus.

1. Der erste Feuerknopfdruck definiert den Startpunkt der Linie (wird durch Blinken deutlich gemacht).
2. Der zweite Knopfdruck zeichnet die Linie (ist zugleich <SHIFT> gedrückt, wird eine vorhandene Linie gelöscht).
3. Mit der <**Pfeil links**>-Taste kann ein markierter Punkt wieder zurückgenommen werden.

<**R**> – Schaltet auf Rechteckmodus.

Die Ausführung geschieht bei <L>.

<**C**> – Schaltet auf Circle-Modus.

Der erste Knopfdruck definiert die Kreismitte, ein zweiter einen beliebigen Punkt auf der Kreislinie.

<**P**> – Schaltet auf Paint-Modus.

Dabei werden auf Knopfdruck Flächen unter dem Cursor gefüllt.

<**B**> – Erhöht die Hintergrundfarbe um 1.

<**F**> – Schaltet auf Grafikfarbmodus.

Zusätzlich wird der Farbmerker um 1 erhöht (der Rahmen zeigt aktuelle Farbe an). Mit Feuer färbt man einen 24 x 21-Punkte-Block, mit <SHIFT F> die gesamte Grafik.

<**SHIFT T**> – Schaltet auf Textfarbemodus.

Der Farbmerker wird um 1 erhöht. Mit Feuer wird ein Zeichen eingefärbt, mit <T> der gesamte Text.

<**SHIFT P**> – Ändert die Text-/Grafikpriorität.

<**T**> – Startet den Texteditor.

In diesem Modus sind alle Funktionen des Basic-Editors aktiv. <F1> führt zurück in den Malmodus.

<**Pfeil oben**> – Text in Grafik

Über die Tastatur läßt sich vierfach großer Text in die Grafik schreiben. Beendet wird die Eingabe mit <RETURN>.

<**CBM S**> – Zum Speichern eines Bildes

Der Textbildschirm wird bei der Filenamenseingabe gerettet (unter ROM).

<**CBM C**> – Gibt den Diskstatus aus.

Läßt die Eingabe von Floppybefehlen zu.

<**CMB P**> – Druckt ein Bild aus.

Ein Auswahlmenü ermöglicht unterschiedliche Arten der Hardcopy.

<**SHIFT X**> – Beendet Hi-Tegra.

Ein Neustart ist jederzeit mit SYS 36864 möglich!

Kurzinfo: Converter

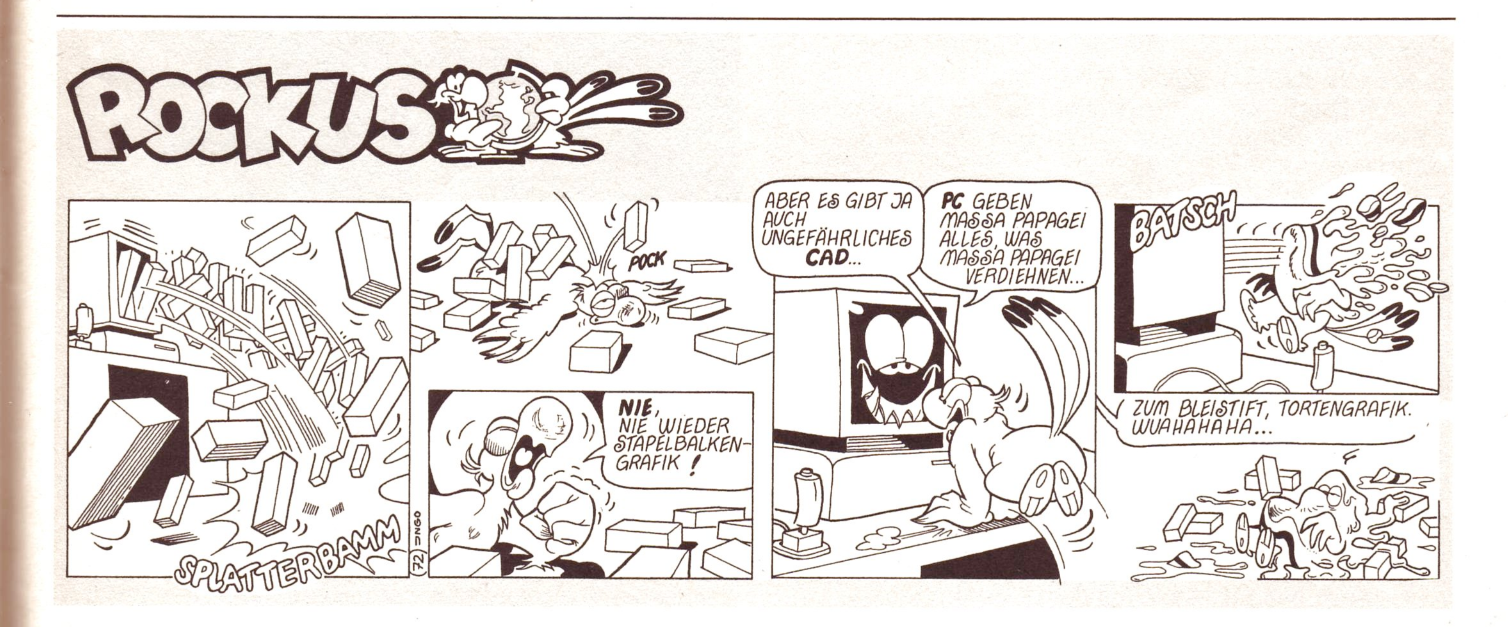
Programmart: Grafikkonvertierprogramm
Laden: LOAD"CONVERTER",8
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur
Besonderheiten: wandelt Hires-Grafik in Tegra-Format
Benötigte Blocks: 3
Programmautor: M. Genter

CONVERTER

Dieses Programm wandelt ein übliches Hires-Bild in eine Tegra-Grafik um. Bei dieser Wandlung werden vier Hires-Punkte zu einem Punkt zusammengefaßt. Das Programm wird mit

```
LOAD"CONVERTER",8
```

geladen und mit RUN gestartet. Danach wird automatisch eine Maschinenroutine nachgeladen. Die Konvertierung eines Bildes dauert ungefähr so lange wie das Laden eines Hires-Bildes (je nach Punkteanzahl bis zu 20 Prozent länger). Auf der beiliegenden Diskette ist eine Weltkarte (Abb. 3) einmal als Hires- und einmal als Tegra-Bild gespeichert. Das Tegra-Bild kann mit TAKE geladen werden, die Hires-Grafik läßt sich z.B. mit Hi-Eddi ansehen. (gr)



von Hartwig zur Nieden

View - Spr

Geister

»Sprite« bedeutet, sinngemäß übersetzt, soviel wie: »Geist, Kobold«. Die Programmierung dieser grafischen Gebilde gerät aber auch manchmal zum »Horrortrip« mit dem Computer. »View« bannt diese »Geister«, die Abwechslung auf eintönige Bildschirme bringen.

Unser »Exorzist« liegt in dreifacher Ausführung auf der Diskette zum Heft vor:
- VIEW-INSTALL, das Grundprogramm. Es dient dazu, den Sprite-Editor an einen von Ihnen gewünschten Speicherbereich anzupassen und zu speichern.
- VIEW BASIC, der bereits installierte Sprite-Editor. Er beginnt hier bei Adresse $0801 (2049, Basic-Anfang) und läßt sich wie ein Basic-Programm laden:

```
LOAD "VIEW BASIC",8
```

Wie gewohnt, können Sie ihn mit RUN starten.
- VIEW 49152, diese Programmversion wird absolut geladen:

```
LOAD "VIEW 49152",8,1
```

und nach vorheriger Eingabe von »NEW« mit »SYS 49152« gestartet.

Für welche Version Sie sich auch entscheiden (Installation in einem gewünschten Speicherbereich mit »VIEW INSTALL« oder Verwendung einer der beiden fertigen Dateien), die Programmfunktionen werden davon nicht beeinträchtigt.

Nach dem Start erscheinen Bildschirm und Rahmen grau. Auf der linken Hälfte des Monitors finden Sie das Muster eines »Riesen-Sprite« (Größe: 24 Zeilen x 21 Spalten, Abb. 1). Rechts daneben befindet sich der »echte« Grafikkobold in Originalgröße. Darunter liegt das noch unbenutzte und daher unsichtbare »Memory-Sprite«, das uns später interessieren soll.

Der obere rechte Bildschirmrand zeigt die Statuszeilen, z.B.:

```
BNK+ BLK+ DIF
0    201  15
12879
```

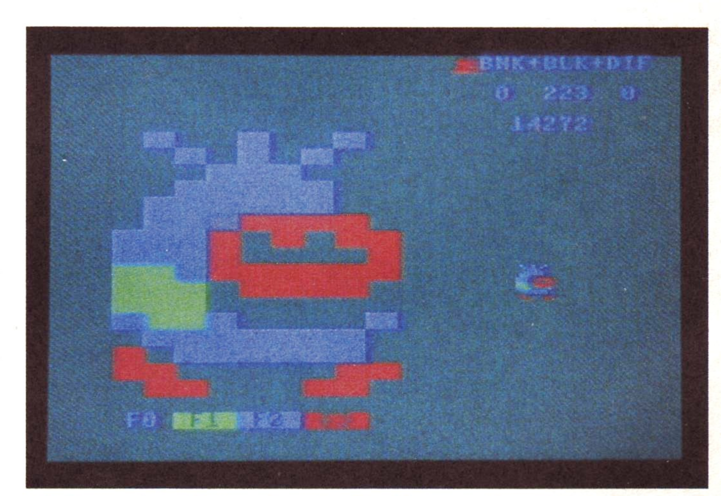
Die Erläuterung der Begriffe:
BNK: bezeichnet die Videobank des VIC-Chips (0 bis 4)
BLK: Sprite-Block in der gewählten Bank (0 bis 255)
DIF: gibt den Byte-Versatz nach oben an, wenn ein Sprite-Muster gelesen wird.

Darunter findet man die entsprechenden Werte (beim Programmstart stehen alle bei »0«). Es folgt die sich daraus ergebende Absolutadresse des Sprite.

Unter dem großen Spritefeld sehen Sie ebenfalls eine Statuszeile mit den aktuellen Multicolorfarben.



[1] Der Suchmodus von »View« mit Riesensprite und Statuszeilen



[2] Bequemes Entwerfen der Sprites per Joystick im Editmodus

SUCHMODUS

1. Joystickfunktionen

»View« besitzt mehrere Modi. Nach dem Start ist der Suchmodus aktiv. Benutzen Sie dazu den Joystick in Port 2:
rechts: ein Sprite-Block (64 Byte) vor,
links: ein Block zurück,
hoch: ein Byte vor,
runter: ein Byte zurück.

Durch Druck auf den Feuerknopf beschleunigt man das Blättern. Sprite-Animationen lassen sich damit komfortabel betrachten.

2. Tastaturfunktionen

Diese sind im Hauptmenü (Suchmodus) und im Save-Modus erreichbar:
<B>: eine Bank (16384 Byte) vor,
<M>: Umschaltung zwischen Hires- und Multicolor-Sprites,
<F1>, <F3>, <F5>, <F7>: Auswahl der Zeichenfarben (nur im Edit-Modus),
<F2>: erhöht die Farbe 0 (Hintergrund),
<F4>: erhöht Farbe 1 (Multicolorfarbe 1)
<F6>: erhöht Farbe 2 (Vordergrund-, Sprite-Farbe),
<F8>: erhöht Farbe 3 (Multicolorfarbe 2),
<E>: vergrößert und verkleinert das Originalsprite in x- und y-Richtung,
<SHIFT CLR/HOME>: löscht das Memory-Sprite (im Editmodus auch das Editierfeld mit dem »Riesen-Sprite«).

EDITMODUS

Weitere Tasten haben für den Editmodus Gültigkeit:
< * >: aktiviert oder beschließt den Editmodus. Die Rahmenfarbe wechselt auf Schwarz. Automatisch wird die Vordergrundfarbe (F3) gewählt und zu Beginn der oberen Statuszeile angezeigt (Abb. 2). Ein Punkt markiert den Cursor im Hires-Modus, bei Multicolor besteht dieser aus zwei Editierpunkten.

te-Editor der Extraklasse

stunde

Im Hires-Modus gibt die Farbnummer an, ob Punkte im Spritemuster gesetzt oder gelöscht werden sollen:
Farbe 2 bedeutet löschen, Farbe 3 setzen.

Bewegt wird der Editor-Cursor mit dem Joystick. Um Punkte zu setzen, muß man den Feuerknopf drücken. Mit den Cursor-Tasten läßt sich das Sprite-Muster auf dem Editorfeld verschieben.

<**X**>, <Y>: spiegelt das Sprite an der x- bzw. y-Achse,
<**R**>: dreht das Sprite im Uhrzeigersinn. Da es sich über 24 x 21 Pixel ausdehnt, wurde darauf geachtet, daß die drei rechten, überstehenden Spalten stehenbleiben. Damit kann man eine Drehbewegung auch rückgängig machen. Bei Multicolor-Sprites werden die unterschiedlichen Farben nicht mitgedreht.

Wenn Sie den Editmodus wieder verlassen (erneuter Druck auf <*>), überträgt das Programm das gerade bearbeitete Sprite-Muster ins Memory-Sprite. Es erscheint nun das vor dem Aufruf des Editmodus aktuelle Sprite, sowohl im Editierfeld als auch verkleinert auf der rechten Seite. Das Memory-Sprite bleibt jetzt so lange im Speicher, bis es mit dem RAM getauscht, überschrieben, gelöscht oder der Load- bzw. Edit-Modus aufgerufen wird. Folgende Tasten übertragen das Memory-Sprite ins RAM:
<**Z**>: tauscht das RAM mit dem Inhalt des Memory-Sprite,
<**SHIFT Z**>: kopiert RAM ins Memory-Sprite,
<**SHIFT CBM Z**>: kopiert das Memory-Sprite ins RAM.

Achtung: Gehen Sie mit der ersten und dritten Tastenfunktion vorsichtig um, da sich damit auch die Zeropage des C64 und das Programm selbst überschreiben läßt. Überprüfen Sie auf jeden Fall vorher die aktuelle Absolutadresse, die unter der oberen Statuszeile gezeigt wird!

ORDNERMODUS

Dazu muß folgende Taste gedrückt werden:
<**O**>: Der Ordnermodus befähigt den Sprite-Editor »View«, in anderen Programmen (z.B. Spielen) auftauchende Figuren zusammenzusetzen, da diese meist aus mehreren Sprites gebildet werden. Ebenso können Sie eigene zusammenhängende Sprites entwerfen, es stehen alle acht zur Verfügung. Rufen Sie diese Funktion zum ersten Mal auf, sind allerdings noch keine Sprites zu sehen. Lediglich am linken unteren Rand findet man folgende Anzeige:

```
0 :0 0
```

Dies entspricht den betreffenden Daten im Format:

```
Spritenummer :X- und Y-Koordinate
```

Die Sprites werden durch den Joystick bewegt, der Feuerknopf beschleunigt diesen Vorgang.
<**SPACE**>: erhöht die Sprite-Nummer (0 bis 7),
<**F1**>: erhöht die Sprite-Farbe.

Den Ordnermodus verläßt man durch erneuten Druck auf <**O**>. Das Programm merkt sich Farben und Koordinatenwerte.

SAVE-MODUS

Um Sprite-Muster auf Diskette zu speichern, muß man in diesen Modus wechseln:
<**S**>: Der Rahmen nimmt die Farbe 2 an, der Bildschirm wird gelöscht. In der linken oberen Ecke auf dem Monitor steht folgende Abfrage:

```
SAVE?
FILENAME :
```

Geben Sie den gewünschten Dateinamen an, unter dem die Spritemuster gespeichert werden sollen. Die Floppystation läuft an, dann kehrt das Programm in den Suchmodus zurück. Die zu speichernden Sprites wählen Sie nun wie gewohnt mit dem Joystick aus. Mit <SPACE> werden sie zur Diskettenstation übertragen. Alle gespeicherten Sprite-Daten erhalten \$2000 (8192) als Ladeadresse.

Ist Ihre Arbeit beendet, verlassen Sie den Save-Modus mit <**RUN/STOP**>. Am oberen Bildschirmrand erscheint der aktuelle Floppystatus, nach Druck auf eine beliebige Taste kommen Sie zurück ins Hauptmenü.

LOAD-MODUS

Folgende Taste aktiviert diesen Programmteil:
<**L**>: In der linken oberen Bildschirmecke steht jetzt diese Meldung:

```
LOAD AB 16384
FILENAME :
```

Der Dateiname muß mit dem übereinstimmen, der beim Speichern gewählt wurde. Die Spritedaten werden ab der Adresse geladen, die **vor** dem Aufruf des Load-Modus Gültigkeit hatte (in unserem Beispiel 16384 = Bank 1, Block 0, Differenz 0). Schon während des Ladevorgangs zeigt der Bildschirm, wie die Sprites aussehen. Ein Druck auf die STOP-Taste bricht den Ladevorgang ab.

Auf der Diskette zum Heft befinden sich drei Beispiel-Sprite-Dateien: *SPR.GARFIELD, SPR.WIZBALL, SPR.BUBBLE*

DOS-MODUS

Sämtliche Diskettenbefehle lassen sich innerhalb des Programmablaufs aktivieren. Drücken Sie folgende Taste, um den Modus einzuschalten:
<@> (At sign): Das Programm fragt Sie nach dem Diskettenbefehl:

```
DOS
FILENAME :
```

Statt eines Dateinamens geben Sie hier den gewünschten DOS-Befehl für die Floppy ein (z.B. »S: Name« zum Löschen einer Datei, »N: Diskname, ID« zum Formatieren einer Diskette usw.). Der Druck auf eine beliebige Taste bringt Sie zurück ins Hauptmenü.

Folgende Tastenfunktionen sind unabhängig vom jeweiligen Modus:
<**D**>: Zeigt das Directory der aktuellen Diskette im Laufwerk an. Bei langen Verzeichnissen hält <SHIFT> die Ausgabe an, <STOP> bricht sie ab. Ein weiterer Tastendruck aktiviert das Hauptmenü.
<**Q**>: Der Sprite-Editor »View« wird ohne Reset verlassen, Sie befinden sich wieder im Direktmodus des C64.

Ob mit eigenen oder aus fremden Programmen »entliehenen« Sprites: Viel Spaß bei der Zusammenstellung aktionreicher Sprite-Animationen! (bl)

Kurzinfo: View

Programmart: Anwendungsprogramm für den Entwurf von Sprites mit Animationssequenzen
Laden: LOAD "VIEW BASIC",8 (für andere Versionen Ladehinweise in der Beschreibung beachten!)
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur/Joystick Port 2
Besonderheiten: Objektcode des Hauptprogramms kann mit »VIEW INSTALL« an beliebige andere Speicherbereiche im Basic-RAM verschoben werden
Benötigte Blocks: 8
Programmautor: Hartwig zur Nieden



FÜR DIE 3 TOP-GRAFIK-LISTINGS ZUM C 64.

Sie programmieren gern oder haben schon interessante Software entwickelt? Halten Sie Ihre Meisterwerke nicht im stillen Kämmerlein zurück. Wir suchen tolle Grafikprogramme und Utilities für den C 64.
Sie haben die Chance, einen Geldpreis bis zu 2000 Mark für so ein Programm zu gewinnen. Selbstverständlich honorieren wir auch jedes andere Listing, das wir veröffentlichen. Schicken Sie Ihre Programme mit Diskette und ausführlicher Beschreibung an:

MARKT & TECHNIK VERLAG AG
Redaktion Sonderhefte
Stichwort: Top-Listing C 64
Hans-Pinsel-Straße 2
8013 Haar bei München

Einsendeschluß: 31.05.91

1. Preis: 2000 DM
2. Preis: 1200 DM
3. Preis: 800 DM

Programmiergrundlagen

Es kommt Bewegung ins Bild

Eine Spezialität des C64 sind »Sprites« – frei definierbare bewegliche Objekte. Ohne großen Aufwand lassen sie sich über den Bildschirm bewegen. Erfahren Sie, wie einfach der Umgang mit Sprites ist.

von Ingolf Koch

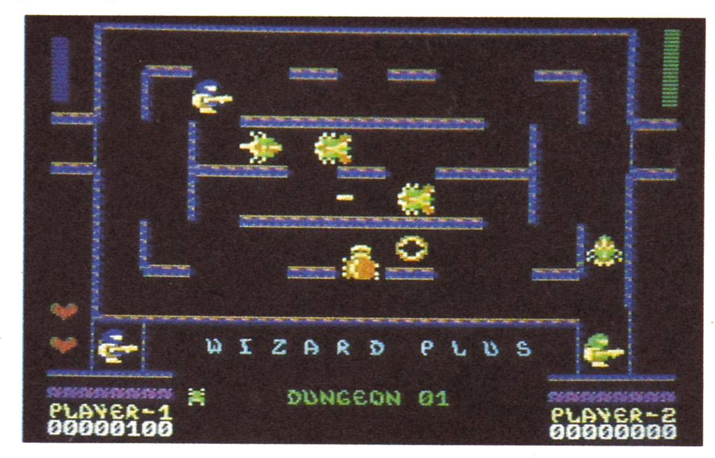
Wie entstehen eigentlich diese kleinen Figuren, die unzähligen Spielen erst ihren Reiz geben? Sprites (sprich »Spraitz«) sind 24 x 21 Punkte große Grafiken, von denen der C64 bis zu acht Stück gleichzeitig auf dem Bildschirm darstellen und unabhängig voneinander bewegen kann (Abb. 1). Dabei ist es nicht wichtig, ob der normale Textmodus, in dem sich der Computer nach dem Einschalten befindet, oder der Grafikmodus (wie in den meisten Spielen) aktiviert ist. Für die Farbgebung können von den 16 Farben des C64 bis zu vier Farben gleichzeitig benutzt werden.

Wie erwähnt, besteht ein Sprite aus einer Grafik mit 24 x 21 Punkten. Es ist in (horizontaler) x-Richtung 24 Grafikpunkte breit und in (vertikaler) y-Richtung 21 Punkte hoch (Abb. 2). Schauen wir uns nun die oberste Zeile dieses Punktfeldes an. Ein Punkt in dieser Zeile kann entweder sichtbar (=1) oder unsichtbar (=0) sein. Das hängt damit zusammen, daß ein Computer prinzipiell nur das Dual- bzw. Binärsystem kennt. Hier gibt es im Gegensatz zum gebräuchlichen Dezimalsystem mit den Ziffern 0 bis 9 nur die beiden Ziffern 0 und 1. Diejenigen, die mit dem Dualsystem noch nicht vertraut sind, finden im Textkasten »Das Binärsystem« eine kurze Einführung in dieses Zahlensystem.

Doch kommen wir auf die erste Zeile unseres Sprite zurück. Um das Aussehen eines Sprite festzulegen, unterteilt man 24 Punkte dieser Zeile in drei Blöcke zu je acht Punkten (Abb. 2). Ein solcher Block wird »Byte« genannt und kann mit dem Basic-Befehl »POKE« in den Speicher, dem Gedächtnis des C64, übertragen werden. Die einzelnen Punkte innerhalb der Bytes tragen den Namen »Bit« und werden nach ihrer Reihenfolge bezeichnet. Dabei ist das Bit, das sich ganz links in einem Byte befindet, das siebte Bit, das zweite von links ist das sechste, und so weiter. Das Bit ganz rechts trägt somit die Nummer 0. Diese Numerierung hängt wieder eng mit dem Dualsystem zusammen. Eine ähnliche existiert bei den Bytes: Das erste in der obersten Zeile ist das Byte 0, das in der Mitte der Zeile das Byte 1, das rechte ist Byte 2. Danach fährt man mit der Numerierung links in der zweiten Zeile von oben fort, bis man schließlich rechts unten bei 62 ankommt. Ein Sprite besteht also aus 64 Byte (Byte 0 mitzuzählen) zu je 8 Bit. Doch wie teilt man dem Computer nun mit, wie das Sprite aussehen soll? Der C64 stellt dafür Sprite-Blöcke zur Verfügung, in die wir unsere Sprites schreiben können.

Diese Blöcke umfassen wieder wegen des Dualsystems (2 x 2 x 2 x 2 x 2 x 2 = 2 hoch 6 = 64Byte) eines mehr als für das Sprite nötig wäre. Nach dem Einschalten des Computers sind nur vier Blöcke für die Sprite-Programmierung nutzbar, und zwar Block 11, 13, 14 und 15.

Die maximal acht Sprites werden Byte für Byte in die Sprite-Blöcke übertragen. Durch das Schreiben der Blocknummer



[1] Typisch für viele Spiele: übers Bild huschende Sprites

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	Dezimalwerte		
Wert / Byte	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1			
0- 2		X	X	X						X	X			X		X	X	X		X	X	X			112	101	220
3- 5	X								X		X		X			X				X			X		128	169	18
6- 8	X							X			X					X	X			X	X	X			129	33	156
9-11	X		X	X			X	X	X	X	X					X				X			X		179	225	18
12-14	X	X			X						X					X	X	X		X				X	200	33	209
15-17	X				X						X														136	32	0
18-20		X	X	X							X														112	32	0
21-23																									0	0	0
24-26																									0	0	0
27-29																									0	0	0
30-32																									0	0	0
33-35																									0	0	0
36-38																									0	0	0
39-41																									0	0	0
42-44																									0	0	0
45-47																									0	0	0
48-50																									0	0	0
51-53																									0	0	0
54-56																									0	0	0
57-59																									0	0	0
60-62																									0	0	0

[2] Der schematische Aufbau unseres Sprite aus Bild 3

Kurzinfo: Demo1

Programmart: Sprite-Demo
Laden: LOAD "DEMO1",8
Starten: nach dem Laden RUN eingeben
Benötigte Blocks: 2
Programmautor: Ingolf Koch

Basis-Adresse des Videochips: 53248	
Register	**Bedeutung**
0	x-Koordinate von Sprite 0
1	y-Koordinate von Sprite 0
2 bis 15	Wie Register 0 und 1 für die Sprites 1 bis 7
16	x-Überlauf: jedem Bit ist das entsprechende Sprite zugeordnet.
21	Sprites ein/aus
23	Sprite vergrößern vertikal
28	Multicolor Modus: Bei gesetztem Bit ist das entsprechende Sprite mehrfarbig dargestellt.
37	Farbe Multicolor 0
38	Farbe Multicolor 1
39	Farbe Sprite 0
40 bis 46	Farbe Sprites 1 bis 7

Tabelle 1. Alle für die Sprite-Programmierung wichtigen Register des Video-Interface-Chips (VIC)

in den jeweiligen Blockzeiger teilen wir dem C64 mit, wo sich die Daten unseres Sprite befinden. Die Zeiger befinden sich zunächst ab Adresse 2040: Der Zeiger auf Sprite 0 liegt in der Speicherzelle 2040+0, der auf Sprite 1 in Adresse 2040 + 1 etc. Ein Beispiel: Liegen die Daten des Sprite 2 in Block 14, teilen Sie es dem Computer mit durch:

```
POKE 2040+2,14
```

Tippen Sie dies ruhig einmal ein - noch passiert nichts.

Wo finden wir nun den Sprite-Block, in den wir noch das Aussehen des Sprite schreiben wollen? Die Anfangsadresse eines Sprite - die Stelle im Speicher, an der sich das erste Byte befindet - errechnet man nach folgender Formel:

Anfangsadresse = Blocknummer x 64.

Da wir Block 14 ausgewählt haben, lautet sie 14 x 64 = 896. Ab jetzt beginnt die eigentliche Arbeit. Dem Computer muß das Sprite Byte für Byte übergeben werden. Das Prinzip ist recht einfach. Man rechnet die einzelnen Bits eines Bytes in Dezimalzahlen um und schreibt sie dann in den Speicher des Computers. Das folgende Beispiel soll diesen Vorgang etwas verdeutlichen. Das Byte 0 des Sprite (oben links in Abb. 2) sieht folgendermaßen aus:

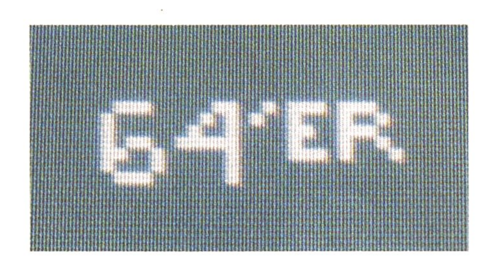
```
76543210 (Nr. des Bits)
..***...
```

Die Zahl für diese Form lautet: 2 hoch 6 + 2 hoch 5 + 2 hoch 4 = 64 + 32 + 16 = 112. Um dieses Byte in den Speicher des Computers zu schreiben, geben Sie

```
POKE 896 + 0,112
```

ein - wieder ein deutlicher Bezug zum Dualsystem!

Wir wissen, wie das Aussehen eines Sprite zu definieren ist. Doch auch bei genauer Einhaltung der Anweisungen erscheint noch kein Sprite auf dem Bildschirm: Es muß erst eingeschaltet werden. Die dafür zuständige Speicherzelle hat die Adresse 53296. Hier setzen Sie nun das dem Sprite ent-



[3] Das Beispiel-Sprite

[4] Ein komfortables Entwurfsblatt für Sprites ▼

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
Wert	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	Dezimalwerte		
Byte																											
0- 2																											
3- 5																											
6- 8																											
9-11																											
12-14																											
15-17																											
18-20																											
21-23																											
24-26																											
27-29																											
30-32																											
33-35																											
36-38																											
39-41																											
42-44																											
45-47																											
48-50																											
51-53																											
54-56																											
57-59																											
60-62																											

sprechende Bit mit:

```
POKE 53296,2 hoch 2
```

um unser Sprite 2 einzuschalten. Jetzt sollten Sie dem Computer nur noch die Stelle des Bildschirms mitteilen, an der Sie Ihr Sprite plazieren wollen.

Der Übersicht halber führen wir noch eine Variable ein, bevor wir weitermachen. Die Basisadresse des VIC – 53248. Das ist die Adresse im Speicher des C64, ab der sich für die Sprite-Steuerung wichtige Speicherstellen (auch »Register«) befinden. Statt der Zahl 53248 wird im folgenden nur noch die Variable »V« auftauchen, was einerseits der Übersichtlichkeit dient und Ihnen andererseits unnötige Tipparbeit erspart. Geben Sie bitte ein:

```
V = 53248
```

Das Register zum Ein- und Ausschalten des Sprite ist Nummer 21 (53248 + 21 = 53269). Statt POKE 53269,n könnten Sie auch POKE V + 21,n schreiben (Abb. 1), wobei es sich bei n um die Nummer des Sprite handelt.

Doch kommen wir zum Positionieren der Sprites auf dem Bildschirm zurück.

Dafür sind zwei Informationen notwendig: Die x- und die y-Koordinate des Sprite auf dem Bildschirm. Bei normaler Bildschirmgrafik sind die Koordinaten 0/0 links oben. Bei Sprites befinden sich diese außerhalb des Text/Grafikfensters. Erst ab Koordinate 24/50 ist ein Sprite vollständig sichtbar. Wenn Sie bei diesen Endwerten beginnen, läßt sich das Sprite komfortabel aus dem sichtbaren Bereich hinein- bzw. hinausbewegen. Geben Sie doch einfach folgende Koordinaten ein:

```
POKE V + 2 x 2,200
```

und

```
POKE V + 2 x 2+1,100
```

Jetzt erscheint ein kleiner Strich auf dem Bildschirm – unser Sprite! Noch ist es nicht komplett, dazu müssen noch die restlichen Zahlen aus der rechten Spalte von Abb. 2 eingegeben werden:

```
POKE 896+1,101
POKE 896+2,220
POKE 896+3,220
POKE 896+4,136
```

bis mindestens zum Byte Nummer 19: Unser Sprite ist fertig (Abb. 3)! Um Ihnen den Entwurf von Sprites zu erleichtern, finden Sie in Abb. 4 eine Konstruktionshilfe.

Doch was wären die Sprites ohne Farbe? Wie eingangs erwähnt, kann man für die Farbgebung die ganze Palette von 16 Farben ausnutzen (Abb. 2). Dabei darf jedes Sprite eine andere Farbe erhalten, die an der Adresse V+39 + Sprite-Nummer gespeichert wird. Der Basic-Befehl dafür lautet

```
POKE V+39 + 2,7
```

um unser Sprite gelb einzufärben. Den Farbcode können Sie der Farbtabelle im Handbuch Ihres C64 entnehmen. Das Wichtigste haben Sie hinter sich. Wenn Sie etwas nicht auf Anhieb verstehen, denken Sie an den Satz »Probieren geht über Studieren«! Probieren Sie ruhig etwas in Basic aus – im schlimmsten Fall »stürzt Ihr Computer ab«, und Sie müssen ihn einmal aus- und wieder einschalten. Zerstören können Sie ihn nicht.

Das Binärsystem

Unser Dezimalsystem umfaßt die Ziffern 0 bis 9. In jeder Stelle sind damit zehn unterschiedliche Werte darstellbar. In der untersten Stelle haben die Zahlen die Wertigkeit:

0,1,2,3,4,5,6,7,8 und 9.

Bei der nächsten Stelle wird die Wertigkeit mal 10 berechnet. Dadurch entstehen:

10,20,30,40,50,60,70,80 und 90

Bei der nächsten Stelle ergibt sich als Rechengrundlage 100 (10x10=100) usw.

Für uns eine selbstverständliche Sache, da wir dieses Dezimalsystem von Kindesbeinen an erlernt haben.

Beim Computer ist diese Art des Rechensystems nicht möglich. Bei den ersten Computern wurden Relais eingesetzt, und die kennen nur den Zustand ein- oder ausgeschaltet. Aber es war möglich, mehrere Relais miteinander zu verknüpfen. Als Ergebnis wurde das Binärsystem eingesetzt. Heute verwendet man freilich keine Relais mehr, aber das Grundprinzip – Strom ein, Strom aus – ist geblieben. Und damit auch das Binärsystem.

Da pro Stelle nur zwei Zustände möglich sind, multipliziert man die nächsthöhere mit dem Wert »2«. Dieses System beruht daher auf der Basis zwei. Mit mehreren Stellen läßt sich wieder jede Zahl ausdrucken. Im folgenden hat in der Binärdarstellung »0« die Bedeutung Strom aus, und »1« bedeutet Strom ein:

Binär	Dezimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Das Beispiel oben zeigt, daß sich mit drei Binärstellen eine Zahlenreihe von 0 bis 7, also acht Zahlen, darstellen lassen. Der C64 hat pro Speicherstelle acht Binärstellen zur Verfügung. Mit diesen läßt sich eine Zahlenreihe von 0 bis 255, also 256 Zahlenwerte, ausdrücken.



SONDER HEFT 64

VORSCHAU

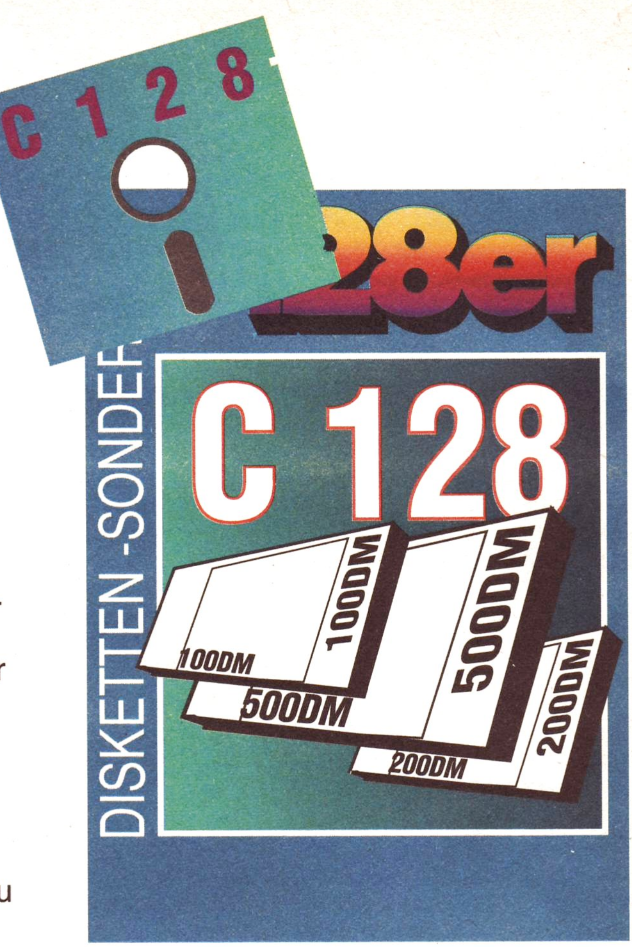
Totgesagte Computer leben länger! Das beweist das nächste Sonderheft, Nr. 64, für die Freunde des C 128.

■ Wir stellen Ihnen die drei Gewinner unseres Programmierwettbewerbs aus dem 128er-Sonderheft 58 vor. Alle Siegerprogramme sind im Heft ausführlich beschrieben und befinden sich auf Diskette.

■ Grafik auf dem 80-Zeichen-Bildschirm, noch dazu mit allen 16 Farben? Laden, Speichern und Drucken einer VDC-Hires-Grafik? Nichts leichter als das! Sämtliche Grafikbefehle des Basic 7.0 gelten jetzt auch im VDC-Modus. Die beiden neuen Versionen von »Graphic-80« halten das, was sie versprechen!

■ Patches, Updates, Tips und Tricks zu neuen und alten C-128-Programmen vermitteln ein »Aha«-Erlebnis nach dem anderen...

■ Ganz besonders widmen wir uns diesmal den viel zu wenig beachteten »Mauerblümchen«: den Dienstprogrammen auf der CP/M 3.0-Diskette.



Das 128'er-Sonderheft 64 gibt's ab 22.3.1991 beim Zeitschriftenhändler

Aus aktuellen oder technischen Gründen können sich Themen ändern. Wir bitten dafür um Verständnis.

Zum Editieren von Sprites befindet sich ein komfortabler Sprite-Editor auf der beiliegenden Diskette. Er ist auf Seite 44 beschrieben.

Jetzt noch Farbe

Bisher konnten wir die Sprites nur als Ganzes durch Änderung ihrer Koordinaten auf dem Bildschirm bewegen. Ihre Form veränderte sich jedoch nicht.

Um einen möglichst realitätsnahen Bewegungsablauf zu simulieren, muß man mehrere ähnliche Sprites entwerfen, die sich z. B. nur durch die Haltung der Arme und Beine unterscheiden, und sie schnell nacheinander auf dem Bildschirm darstellen (Filmeffekt). Diesen, dem Daumenkino nachempfundenen, Vorgang nennt man Animation. Zur Verdeutlichung dient folgendes Beispiel:

Eine Bewegung soll in vier Bildern gespeichert werden und anschließend als »Film« auf dem Bildschirm erscheinen. Man legt dafür die Bilder in vier Blöcken ab und schaltet ein Sprite wie oben erklärt an. Dabei zeigt der Blockzeiger auf den ersten der benutzten Blöcke. Im weiteren Ablauf des Programms wird dieser Zeiger verändert (auf den 2., den 3., ... der benutzten Blöcke). So entsteht der Eindruck, das Sprite bewege sich in sich selbst. Laden Sie das Demoprogramm von der beiliegenden Diskette mit:

```
LOAD"DEMO1",8
```

und starten Sie es mit RUN. Nach kurzer Zeit erscheinen ein Frachter und ein Fisch auf dem Bildschirm. Zur Erklärung des Programms müssen wir weiter ausholen: Leider hat man nach dem Einschalten des Computers nur für vier Sprite-Definitionen Speicherplatz, was besonders für animierte Sprites nicht ausreicht. Abhilfe schafft man sich, indem man das Ende des Basic-Speichers durch POKE 56,128 etwas niedriger legt. Dadurch haben Sie genug Speicherplatz für Sprites reserviert. Das müssen Sie natürlich dem Computer mitteilen. Zu berücksichtigen ist dabei, daß der VIC nur einen Speicherbereich von 16 KByte ansprechen (adressieren) kann, während der gesamte Speicher des C 64 viermal soviel umfaßt. Numeriert man diese 16-K-Blöcke von 0 bis 3 durch, kann man zwischen ihnen furch den Befehl

```
POKE 56576,3 - Blocknummer
```

umschalten. Durch

```
POKE 56576,1
```

schaltet man den VIC auf den Speicherbereich 2 (von Byte 32768 bis Byte 49151) um. Nun teilen wir noch dem Betriebssystem des Computers mit, daß der Bildschirmspeicher jetzt auch in diesem Speicherbereich liegt. Das geschieht mit dem Befehl

```
POKE 648,132
```

Der Bildschirmspeicher liegt jetzt ab der Adresse 33792 im Speicher. Die Sprite-Zeiger befinden sich nicht mehr an den Adressen 2040 bis 2047, sondern 34808 bis 34815. Mit Hilfe dieser Umschaltung kann man die Blöcke 0 bis 15, 32 bis 63 und 128 bis 255 benutzen. Wichtig ist, daß man beim Errechnen der Anfangsadresse der Sprite-Blöcke nicht mehr Adresse = Blocknummer x 64 benutzt. Zum Ergebnis muß 32768 addiert werden. (gr)

SUCHE: GEOS-DISKETTE,
BELIEBIGE VERSION.
BIETE: 40 MARK BONUS.

UPGRADE: Aus alt mach' neu

Denken Sie doch einmal scharf nach: Verstaubt bei Ihnen zu Hause eine Diskette? Wahrscheinlich mehr als eine. Ist darunter auch eine von Commodore? Zufällig mit dem Aufkleber »GEOS«? Hervorragend. Diese Diskette ist nämlich mehr wert, als Sie bislang gedacht haben: immerhin 40 Mark. Vierzig Mark, die Sie vielleicht schon wegwerfen wollten.

Ein tolles Angebot.

Wir haben ein exklusives Angebot. Sie überlassen uns die alte GEOS-Diskette. **Wir geben Ihnen das Beste und Neueste, was wir anzubieten haben: GEOS 2.0,** die neue Dimension. Sie haben sicher schon davon gehört. Ansonsten lesen Sie doch an anderer Stelle in diesem Heft nach, warum GEOS 2.0 so einzigartig ist: weil es neue Leistungsmaßstäbe setzt, aber ganz einfach zu bedienen ist. Für andere Leute hat dies seinen Preis. Aber nicht für Sie. Sie haben schon ein GEOS; das soll sich jetzt auszahlen. **Sie sparen 40 Mark und bekommen vier neue Programmdisketten, randvoll mit Software.** Dazu natürlich eine komplette Dokumentation.



Und was Sie bisher mit GEOS realisiert haben, läßt sich problemlos weiterverwenden.

Nehmen Sie also Ihre alte GEOS-Diskette. Die Version spielt keine Rolle: **GEOS 1.2, 1.3 oder 128,** das ist unerheblich. Von Commodore oder nicht, danach fragt niemand. Nicht einmal das Handbuch brauchen wir – behalten Sie es. Es kommt uns wirklich nur auf die Diskette mit dem Original-Aufkleber an. Füllen Sie dann den Upgrade-Coupon aus. Schreiben Sie einen Verrechnungsscheck über **49 DM für GEOS 64 2.0 oder 79 DM für GEOS 128 2.0.** Das war's schon. Schicken Sie uns jetzt Diskette, Coupon und Scheck. Sie bekommen dann GEOS 2.0. (Update von 64er auf 128er nicht möglich).
Jeder will es haben – Sie bekommen es zum Sonderpreis. Nehmen Sie diese Gelegenheit wahr, am besten gleich.
Ein Tip zum Weitersagen!

Berkeley Softworks

Berkeley Softworks – Kluge Köpfe setzen auf GEOS 2.0.

UPGRADE-COUPON

Bitte senden Sie mir

- ☐ das Update auf GEOS 2.0 für den C64 für DM 49,–, Bestell-Nr. 51677U
- ☐ das Update auf GEOS 2.0 für den C128 für DM 79,–, Bestell-Nr. 51683U
- ☐ weiteres Informationsmaterial über GEOS 2.0 W 164
- ☐ Originaldiskette liegt bei ☐ Verrechnungsscheck liegt bei
- ☐ Überweisung erfolgt per Zahlkarte nur auf Postgiro-Kto. 14199-803 PA München

Name ____________________

Straße ____________________

PLZ/Ort ____________________

Bitte ausschneiden und schicken an: Markt&Technik Verlag AG, Buch- und Software-Verlag, Kundenbetreuung, Hans-Pinsel-Str. 2, 8013 Haar b. München

2304/009



Markt&Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften und in den Fachabteilungen der Warenhäuser